**REST API - Version 1**

**Authentication**

API requests are authenticated using the [OAuth 2](http://oauth.net/2/) authentication token mechanisms. Every API request MUST contain an "access\_token" parameter. Access tokens can be obtained using one of the following flows:

**Username and Password Authentication**

Third party applications may exchange a user's credentials to obtain an access token on behalf of that user. Send a request to the URL that grants access tokens with grant\_type set to "password". Include the user's username and password as parameters. Example: /api/v1/access-token/?grant\_type=password&username=jdoe&password=jdoe

Use this authentication method for applications that users trust to handle their credentials.

**Session Authentication**

Users logged in to LifeSize Video Center can obtain access tokens using the login session. Access the URL that grants access tokens from a browser where you are logged in to Video Center. Set the grant\_type parameter to "session". Example: [/api/v1/access-token/?grant\_type=session](https://172.29.28.54/api/v1/access-token/?grant_type=session)

Use this authentication method for simple command line scripts and for exploring the API using a web browser. Users can log in to LifeSize Video Center, obtain a token using the browser, and use the token with command line utilities such as curl until the token expires. Examples on this page use a token generated from the current session.

**Token Validity and Throttling**

API access tokens generated using any of the authentication methods are usually valid for 24 hours. The validity of a token is indicated by the expires\_in parameter in the token granting request's response. The expires\_in parameter's value is in seconds.

A user cannot generate more than 1000 API access tokens over a 24 hour period. Applications that send a large number of API requests should not generate a token for each request. Generate a token and use it for all subsequent requests until it is valid to avoid running into this limit.

**Read API**

Use the HTTP GET method to obtain listings of resources such as users, groups, channels, recording keys and videos on LifeSize Video Center. Each resource type has a list URL that can be used to retrieve a listing of all objects. Object listings show important properties of the listed objects. Each object in the list has a "resource\_uri" attribute that can be used to query all properties of the object.

The following query parameters are supported in GET requests to all resources:

|  |  |
| --- | --- |
| format | Specifies the format of the returned data. Default format is "json." Other supported values are "xml". |
| fields | Specifies the properties required in the response. Property names should be separated by commas. Omitting this parameter returns all properties of the object when a specific object is queried. Queries on list URLs return a limited set of important fields when this parameter is not used. |
| offset, limit | List URLs support paging of results. Use "offset" to specify the index (zero based) of the starting object in the result and "limit" to specify the number of items to be returned. The result contains a "count" attribute that contains the total number of objects on the server. Offset defaults to 0 and limit defaults to 20 if no values are specified. |
| order\_by | Specifies the parameter to be used to sort results returned by list URLs. The default sort order is the creation date of the resource (newer objects first). Prefix field name with "-" to reverse the sort order. |
| filtering parameters | Requests to list URLs may specify filters based on object properties. This parameter follows the [Django query filter syntax](http://docs.djangoproject.com/en/dev/topics/db/queries/#field-lookups). Refer to the video listing examples for details. |

**Create API**

Create new objects on LifeSize Video Center using HTTP POST requests to the resource list URL. The POST request should contain form data (content-type header set to application/x-www-form-urlencoded) specifying the object properties. The properties of the object that is created are returned in the response. The format parameter can be used to specify the response format (default is JSON). The response will contain the "resource\_uri" parameter that can be used to perform subsequent operations on the object. There are restrictions on the type of resources that a user can create. To create resources such as users, groups, and channels, the user must have an access token that has administration privileges. Recording keys can be created only by administrators and content creators who are allowed to create keys. Videos can be uploaded by administrators and content creators using the API.

**Update API**

Update properties of existing resources using HTTP PUT requests to the resource URL. The request body should contain form data (content-type header set to application/x-www-form-urlencoded). Properties in the form data will overwrite corresponding properties of the resource. An object's properties can be updated only by the owner or an administrator.

**Delete API**

Delete an existing resource by sending an HTTP DELETE method to the object's URL. An object can be deleted only by the owner or an administrator.

**Examples**

**Videos**

* List all videos: [/api/v1/recordings?access\_token=e08e1cc4aca84c67](https://172.29.28.54/api/v1/recordings?access_token=e08e1cc4aca84c67)
* Properties of a specific video: [/api/v1/recordings/3603?access\_token=e08e1cc4aca84c67](https://172.29.28.54/api/v1/recordings/3603?access_token=e08e1cc4aca84c67)
* Properties in XML format: [/api/v1/recordings/3603?format=xml&access\_token=e08e1cc4aca84c67](https://172.29.28.54/api/v1/recordings/3603?format=xml&access_token=e08e1cc4aca84c67)
* List the 10 most recent videos with specific properties in the list: [/api/v1/recordings?limit=10&offset=0&fields=recordingfeed\_set,rtmp\_streamer&access\_token=e08e1cc4aca84c67](https://172.29.28.54/api/v1/recordings?limit=10&offset=0&fields=recordingfeed_set,rtmp_streamer&access_token=e08e1cc4aca84c67)
* All videos created after a specific date: [/api/v1/recordings?limit=10&offset=0&date\_created\_\_gte=2010-11-15+08:00:00&access\_token=e08e1cc4aca84c67](https://172.29.28.54/api/v1/recordings?limit=10&offset=0&date_created__gte=2010-11-15+08:00:00&access_token=e08e1cc4aca84c67)
* Videos owned by a specific user: [/api/v1/recordings?limit=10&offset=0&author=281&access\_token=e08e1cc4aca84c67](https://172.29.28.54/api/v1/recordings?limit=10&offset=0&author=281&access_token=e08e1cc4aca84c67)
* Uploading a video file (using curl as client):

curl -k -X POST --form "name=Uploaded using API" --form "description=Video uploaded using API" --form file="@/home/binu/small-movie.mp4" --form "channel=3" --form "author=110" --form "viewers=user1,user2" --form "viewer\_groups=group1,group2" --form "is\_public=false" https://172.29.28.54/api/v1/recordings\?access\_token=e08e1cc4aca84c67

This command instructs curl to sumbit a form with encoding type "multipart/form-data". The name of the recording has been specified and the file is uploaded with the parameter name "file." Additional properties such as tags and description may also be specified during the upload. If the author is not specified, the video will belong to the user uploading the video. Administrators may specify the author by using the ID of the author. The channel field should use the ID of the channel (defaults to the main channel if unspecified). The list of viewers and groups allowed to watch the video can be specified using usernames or names of groups.

* Updating a video's name and tags:

curl -k -X PUT -d "name=Edited+With+API&tags=api+edited" https://172.29.28.54/api/v1/recordings/3603\?access\_token=e08e1cc4aca84c67

This command instructs curl to sumbit a form using the PUT method. It updates the name and the tags of the video. Video update API can be used by administrators only.

* Deleting a video:

curl -k -X DELETE https://172.29.28.54/api/v1/recordings/3603\?access\_token=e08e1cc4aca84c67

This command instructs curl to send a request using the DELETE method on the video's URL. Video deletion API can be used by administrators only.

**Live Content**

* List current live streams: [/api/v1/live-streams?access\_token=e08e1cc4aca84c67](https://172.29.28.54/api/v1/live-streams?access_token=e08e1cc4aca84c67)
* Response fields are similar to the recording API.

**Calls**

* List current calls: [/api/v1/calls?access\_token=e08e1cc4aca84c67](https://172.29.28.54/api/v1/calls?access_token=e08e1cc4aca84c67)

**Channels**

* List all channels: [/api/v1/channels?access\_token=e08e1cc4aca84c67](https://172.29.28.54/api/v1/channels?access_token=e08e1cc4aca84c67)
* Properties of a specific channel: [/api/v1/channels/81?access\_token=e08e1cc4aca84c67](https://172.29.28.54/api/v1/channels/81?access_token=e08e1cc4aca84c67)

**Users**

* List all users: [/api/v1/users?access\_token=e08e1cc4aca84c67](https://172.29.28.54/api/v1/users?access_token=e08e1cc4aca84c67)
* Properties of a specific user: [/api/v1/users/281?access\_token=e08e1cc4aca84c67](https://172.29.28.54/api/v1/users/281?access_token=e08e1cc4aca84c67)
* Creating a new user with recording permissions (default recording bitrate at 768 kb/s, other properties set to global recording defaults):

curl -k -X POST --form "username=newuser" --form "password=password" --form "privileges=administrator:0,can\_record:1,default\_recording\_quality:768" https://172.29.28.54/api/v1/users\?access\_token=e08e1cc4aca84c67

* Importing an LDAP user and assigning recording permissions:

curl -k -X POST --form "import\_ldap=true" --form "username=newuser" --form "privileges=can\_record:1" https://172.29.28.54/api/v1/users\?access\_token=e08e1cc4aca84c67

**Groups**

* List all groups: [/api/v1/groups?access\_token=e08e1cc4aca84c67](https://172.29.28.54/api/v1/groups?access_token=e08e1cc4aca84c67)
* Properties of a specific group: [/api/v1/groups/1?access\_token=e08e1cc4aca84c67](https://172.29.28.54/api/v1/groups/1?access_token=e08e1cc4aca84c67)
* Creating a local group with users for server administration:

curl -k -X POST --form "name=server-administrators" --form "privileges=administrator:1,can\_record:1" --form "users=username1,username2" https://172.29.28.54/api/v1/groups\?access\_token=e08e1cc4aca84c67

* Importing an LDAP group and assigning recording permissions to it:

curl -k -X POST --form "import\_ldap=true" --form "name=Marketing Group" --form "privileges=can\_record:1" https://172.29.28.54/api/v1/groups\?access\_token=e08e1cc4aca84c67

**Recording Keys**

* List all recording keys: [/api/v1/recording-keys?access\_token=e08e1cc4aca84c67](https://172.29.28.54/api/v1/recording-keys?access_token=e08e1cc4aca84c67)
* Properties of a specific key: [/api/v1/recording-keys/251?access\_token=e08e1cc4aca84c67](https://172.29.28.54/api/v1/recording-keys/251?access_token=e08e1cc4aca84c67)
* Reserved Urls: [/api/v1/get-reserved-urls/251?access\_token=e08e1cc4aca84c67](https://172.29.28.54/api/v1/get-reserved-urls/251?access_token=e08e1cc4aca84c67)
* Non admin users will not see keys that are not owned by them.
* Creation and editing of keys is not supported.
* Administrators can delete keys.

**Video Search**

* List all recordings matching search query 'test': [/api/v1/search?q=test&access\_token=e08e1cc4aca84c67](https://172.29.28.54/api/v1/search?q=test&access_token=e08e1cc4aca84c67)
* Search query with other fields: [/api/v1/search?q=test&limit=10&offset=0&fields=name,description,id&access\_token=e08e1cc4aca84c67](https://172.29.28.54/api/v1/search?q=test&limit=10&offset=0&fields=name,description,id&access_token=e08e1cc4aca84c67)
* List all recordings matching search query 'test' and username 'newuser': [/api/v1/search?q=test&user=newuser&access\_token=e08e1cc4aca84c67](https://172.29.28.54/api/v1/search?q=test&user=newuser&access_token=e08e1cc4aca84c67)

**License Type**

* Check license edition used by Video Center: [/api/v1/license-type?app=VC&access\_token=e08e1cc4aca84c67](https://172.29.28.54/api/v1/license-type?app=VC&access_token=e08e1cc4aca84c67)
* Check if Video Center has an Enterprise license: [/api/v1/license-type?app=VC&check=enterprise&access\_token=e08e1cc4aca84c67](https://172.29.28.54/api/v1/license-type?app=VC&check=enterprise&access_token=e08e1cc4aca84c67)

**SIP Dialout**

SIP dialout recordings can be initiated with an HTTP POST request containing the recording parameters as a JSON object. The owner of the recording will be the user initiating the API request. Administrators can specify a different owner by adding an 'author' parameter that contains the username of the owner. This API requires content creation permissions.

* Initiating a SIP dialout recording for author 'newuser':
* curl -k -H "Content-Type: application/json" -H "Accept: application/json" -X POST -d '{"author":"newuser","pin":123,"bitrate\_kbps":512,"channel":"Main","name":"Apivideo","url":"10.95.17.175","is\_recorded":true,"is\_live":true}' https://172.29.28.54/api/v1/dialout?access\_token=e08e1cc4aca84c67

* Creating a SIP dialout recording owned by the current user:
* curl -k -H "Content-Type: application/json" -H "Accept: application/json" -X POST -d '{"pin":123,"bitrate\_kbps":512,"channel":"Main","name":"Apivideo","url":"10.95.17.175","is\_recorded":true,"is\_live":true}' https://172.29.28.54/api/v1/dialout?access\_token=e08e1cc4aca84c67

* Creating a SIP dialout recording owned by the current user without sharing with all users and using 'is\_public' parameter by setting it to false:
* curl -k -H "Content-Type: application/json" -H "Accept: application/json" -X POST -d '{"pin":123,"bitrate\_kbps":512,"channel":"Main","name":"Apivideo","url":"10.95.17.175","is\_recorded":true,"is\_live":true,"is\_public":false}' https://172.29.28.54/api/v1/dialout?access\_token=e08e1cc4aca84c67

Top of Form



Language: 

Bottom of Form