



Developer Guide

Developer Guide

Document Publication Date: March 28, 2016

Copyright © 2009-2015 BroadVision, Inc., 1700 Seaport Boulevard, Suite 210, Redwood City, California 94063, U.S.A. All rights reserved.

The software and accompanying written materials are protected by U.S. and International copyright law. Unauthorized copying of the software, including software that has been modified, merged, or included with other software, or other written material is expressly forbidden. This software is provided under the terms of a license between BroadVision and the recipient, and its use is subject to the terms of that license. Recipient may be held legally responsible for any copyright infringement that is caused or incurred by recipient's failure to abide by the terms of the license agreement.

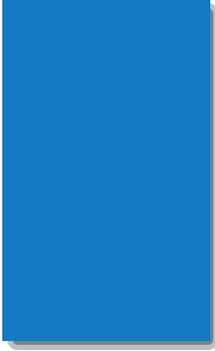
RESTRICTED RIGHTS LEGEND: Terms and Conditions Applicable to Federal Governmental End Users. BroadVision licenses products for ultimate end use by federal government agencies and other federal government customers ("federal government customers") only under the following conditions. Software and technical data rights in these products include only those rights customarily provided to end use customers of Software as defined in the BroadVision Master Software License and Services Agreement and any exhibit thereto. This customary commercial license in technical data and software is provided in accordance with FAR 12.211 (Technical Data) and 12.212 (Computer Software) and, for Department of Defense purchases, DFAR 252.227-7015 (Technical Data - Commercial Items) and DFAR 227.7202-3 (Rights in Commercial Computer Software or Computer Software Documentation). If a federal government or other public sector customer has a need for rights not conveyed under these terms, it must negotiate with BroadVision to determine if there are acceptable terms for transferring such rights, and a mutually acceptable written agreement specifically conveying such rights must be executed by both parties.

The product described in this manual may be protected by one or more U.S. and International patents. Certain applications of BroadVision One-To-One® software are covered by U.S. patent 5,710,887.

DISCLAIMER: BroadVision, Inc., makes no representations or warranties with respect to the contents or use of this publication. Further, BroadVision, Inc., reserves the right to revise this publication and to make changes in its contents at any time, without obligation to notify any person or entity of such revisions or changes.

TRADEMARKS: BroadVision, Clearvale, and the Clearvale logo are trademarks or registered trademarks of BroadVision, Inc., in the United States and other countries.

All other trademarks, service marks, and trade names belong to their respective owners. BroadVision, Inc., disclaims any proprietary interest in the marks and names of others.



Contents

	About This Book	5
	Audience	5
	Organization	5
	Typographical Conventions	6
1	Introduction to Clearvale	7
	Features Overview	8
	Terms Used in Clearvale	11
2	APIs	13
	Overview	14
	Request Format	14
	Response Format	15
	Using the API	16
	Activity	17
	Analytics	33
	Blogs	34
	Bookmarks	53
	Calendar	59
	Communities	67
	CVAuth	78
	Documents	85
	Forums	119
	Membership	130
	Messages	145
	Microblog	147
	Org	149
	Page	156
	Poll	164
	Profile	171
	Search	179
	Statistics	181
	Task	184
3	Customizing Clearvale	213
	Signing Into Clearvale Through Another Application	213
	Configuring Your Network to Use a Different Login Page	214
	Editing Your Login Page to Authenticate for Clearvale	214

Troubleshooting 215

4 External Widgets 217

What Are External Widgets? 217

Creating an External Widget 218

 Downloading a Provider Token From BroadVision 219

 Creating a Web Application for Your External Widget 220

 Creating an External Widget Manifest File 230

Installing Your External Widget 231

 Adding Your Widget to the Account Library 232

 Adding Your Widget to the Global Library 233

 Approving Global Library Widgets 234

Installing a Widget from the Account or Global Library into Your Network 235

Configuring an External Widget 236

Removing External Widgets 238

Index 239



About This Book

This book describes the APIs you can use to access and work with Clearvale[®] features. It also covers ways to customize the Clearvale experience.

Audience

The *Clearvale Developer Guide* is for developers with experience using HTTP and HTTPS protocols, the REST architecture, PHP, and data formats such as JSON or XML. Note that to create external widgets, you must be able to create a small web application, for example in JavaScript or PHP.

Organization

This book is organized as follows:

- [Chapter 1, “Introduction to Clearvale,”](#) gives an overview of Clearvale.
- [Chapter 2, “APIs,”](#) describes the APIs and provides examples of how to use them.
- [Chapter 3, “Customizing Clearvale,”](#) covers how to use the login page of another application to also log into Clearvale.
- [Chapter 4, “External Widgets,”](#) describes how to create external widgets for use in your Clearvale network. It also covers publishing your external widgets so that other Clearvale networks can use them.

Typographical Conventions

This manual incorporates these typographical conventions.

Fonts

- Sample code, commands, filenames, and directory names, are shown in *monospace font*:

```
options='type=file:filetype=image:sort=+time_created'
```

- Variables for which you must provide information are shown in *italic monospace font*:

```
http://network_url/pg/cvapi/rest/format
```

- User interface menu items, buttons, and links that you are instructed to click are shown in bold.

File paths

- By default, file paths are shown in Unix format. For example:

```
directory/file_name
```

The equivalent on Windows would be:

```
directory\file_name
```

Special notes

NOTE: This lead-in calls out helpful information that is related to but distinct from the main topic.

TIP: This lead-in indicates additional details, alternative methods, hints, shortcuts, and related information about the main topic.

IMPORTANT: This lead-in indicates situations that can prevent the system from running and how to avoid them.

1

Introduction to Clearvale

This chapter contains the following topics:

- “Features Overview,” next
- “Terms Used in Clearvale” on page 11

Features Overview

Clearvale is a cloud-based enterprise social network that provides easy and secure collaboration between employees, partners, and customers. You can limit your network to just people within your company. Or, you can set up your network to include people outside of your company, such as partners, customers, or even the general public.

Specific Clearvale features that you can access by using the Clearvale API include the following:

Feature	Description	APIs to Use
Activities	As you and other network members participate in a network, an activity stream is created that tracks the activity happening in your network. In addition to creating an activity stream, network members accumulate activity points for participating in the network.	Activity APIs. For example, get the activity stream for a specific user, get the activity points for a particular user, or get the activity stream for a community.
Blogs	Write and keep your own blog space updated, and leave your feedback and comments for others. Share breaking company or product news, ideas, and more.	Blogs API. For example, get a list of blog folders created by a specific user.
Bookmarks	Maintain links to pages in your Clearvale network or to web pages outside of your network.	Bookmarks API. For example, get a list of all bookmarks created by a specific user.
Communities	Create communities around departments, projects or general interests and organize information in one central location, encouraging collaboration and interaction between your team members. Contextualize all the information for your projects and have it readily available in an instantly accessible space.	Communities API. For example, get a list of documents in a given category within a community.
Contacts	View a full, alphabetized list of your contacts for quick and easy communication.	Profile API. For example, get a list of contacts for a specific user.
Files	<p>Upload files (documents, photos, videos, embedded content) and share them with individuals, communities, or networks. Optionally associate files with a folder or with multiple folders and display them in widgets. Access different versions of files and get history information.</p> <p>Clearvale has three kinds of spaces for folders and folder hierarchies: A network, a community, and a personal space. In each type, the folder hierarchy can contain top folders and subfolders. Each space has a special folder named “uncategorized,” which holds files available in the space and that have not been added to any folders. Network administrators must manage the network folders. Community or network administrators must manage community folders, and personal folders can be managed by the file owners or network administrators.</p> <p>A file can only be created in a personal space, but it can be shared to other spaces (including personal, community and network spaces) for viewing or editing. Once a file is shared to other spaces, it can also be added to folders in those spaces.</p> <p>To add a file to a folder requires that the user have the admin role for that folder and have access to the file.</p>	<p>Communities API. For example, get a list of documents in a given category within a community.</p> <p>Documents API. For example, update a document; get document history.</p>
Forums	Brainstorm and discuss ideas for projects, or provide support for one-time and ongoing issues.	Forums API. For example, create a new forum topic for a specific forum.

Feature	Description	APIs to Use
Members and Member Profiles	<p>Keep your own individual, customizable profile page up to date with your work and contact information.</p> <p>Search for other network members based upon their departments, interests, and more.</p>	<p>Blogs API. For example, get a list of blogs created by a specific user.</p> <p>CVAuth API. For example, authenticate a specified username and password.</p> <p>Forums API. For example, delete forum comments by a user.</p> <p>Membership API. For example, remove a member.</p> <p>Profile API. For example, update a profile for a specified user.</p>
Polls	Gather feedback from network or community members by conducting polls. Conduct polls on the Network page, within a specific community, or on a personal page.	Poll API. For example, get the results of a specific poll.
Tasks	Manage corporate activities, such as document approval, by using tasks. When you upload a document, create a task of approving the document and assign the task to one or more people with approval authority.	Task API. For example, get all tasks that are assigned (open) for a specific user.
Wiki Pages	Create wiki pages of text in your network. Wiki pages are useful for content that you update frequently.	Page API. For example, create new wiki pages of text.

With Clearvale's API, developers can integrate Clearvale features into existing websites and desktop clients. Other ways in which you may want to use the APIs include:

- Collecting data about what content your customers find most useful. For example, you might want to know if customers find the new datasheets on your site helpful.
- Determining which members are most active. You might need to know which of your customers is participating the most in your customer service extranet site.
- Giving users the ability to comment on Clearvale blogs from another website. You might want to make blog posts visible to people beyond your primary Clearvale network—perhaps on an intranet site that deals with customer issues.
- Adding groups of members to a Clearvale community. Perhaps you recently made an organization change and now need to add a large number of members to newly created community; with the APIs, you can do this.
- Authenticating members. For example, you might want members on your existing website to have access to a Clearvale network. Use the CVAuth API to authenticate members.
- Searching for documents from another site. You might need to search for some documents that reside in Clearvale from another intranet site.

Because they are based on the RESTful architecture, whatever you use the APIs for, you will find them easy to use.

Terms Used in Clearvale

The following table lists terms commonly used when working with Clearvale.

Term	Definition
Administrator	Administrator is a role assigned to a network member by the network owner. Administrators have full management control over a network and can manage users, change all network administrative settings, and moderate content. Administrators may assign roles to members but cannot promote members to be Administrators, demote Administrators to other roles, or transfer ownership.
Community	Communities are groups to which members can belong that revolve around similar interests. Communities are useful for organizing information for projects in a central location, encouraging collaboration between team members, and controlling access to sensitive information.
Ecosystem	An ecosystem is a collection of networks. An administrative console is provided for an ecosystem. Ecosystems are available as a premium service.
Members	Members are Clearvale network users. Members are assigned roles that determine what actions they can perform on the network.
Owner	Owner is a role that is automatically assigned to the creator of a network. Owners assign roles to network members and can transfer ownership to another member. Only an owner can disable a network.
Roles	Roles are assigned by network owners or by ecosystem administrators to each network member. The roles are: owner, administrator, moderator, writer, and reader.
Widgets	Widgets are blocks that members with appropriate permissions can place on a Network page, community page, or their own profiles. These blocks can contain file information, photos, videos, and so forth.

The next chapter describes the API.

Chapter 1 Introduction to Clearvale
Terms Used in Clearvale

2 APIs

The Clearvale API is a collection of REST API methods that allows developers to integrate Clearvale or specific features of Clearvale with an existing application.

This chapter contains the following topics:

- “[Overview](#),” [next](#) gives an overview of the API request and response formats.
- “[Using the API](#)” [on page 16](#), lists the API methods and provides descriptions of each one.

Overview

The API functionalities are packaged in a service called Clearvale API. The service is disabled by default when a new network is established. Your BroadVision sales representative can enable the Clearvale API service for your network.

Request Format

The URL endpoint format is `http://network_url/pg/cvapi/rest/format`, where `network_url` is the network address and `format` indicates the [Response Format](#) of the response data. Note that both HTTP and HTTPS are supported, but that when passing passwords, HTTPS is required.

Method parameters are passed as part of an HTTP GET request or as part of the HTTP POST request, depending on the type of method. The Clearvale API has three global parameters for all methods:

- `method`. This parameter specifies which Clearvale API method to execute.
- `auth_token`. This parameter is available for all methods with the exception of `cvauth.authenticate`. The syntax is `auth_token=token`. Tokens are stored on the server and expire two hours after first being issued. The error code for expired tokens is `-1`. This parameter is required unless mentioned otherwise. In cases where it is optional, if you omit it, Clearvale assumes that you are making an anonymous call.
- `jsoncallback`. A parameter used only with JSON endpoints. When present, the return data will be in a JSONP format (`callback({})`), where `callback` is the value of `jsoncallback`. If this parameter is not present, a JSON format is used (`{...}`).

URL encoding All request parameter values must be URL-encoded. Note that URL encoding is language specific, so choose an approach appropriate for your language.

Request examples For a GET request, your URL could resemble the following:

```
http://network_url/pg/cvapi/rest/xml?method=analytics.  
get_activity_points&auth_token=df202eb08cfaf08f61c6f987b0d68e5e&user_id=36
```

or

```
http://network_url/pg/cvapi/rest/json?method=analytics.  
get_activity_points&auth_token=df202eb08cfaf08f61c6f987b0d68e5e&user_id=36
```

If you call a method that uses a POST request, POST requests are language-specific and the approach will vary.

For example, a POST request might look something like this:

```
...  
<form action="http://network_url/pg/cvapi/rest/xml" method="POST">  
<input type="hidden" name="method" value="..." />  
<input type="hidden" name="auth_token" value="..." />
```

More input fields

```
<input type="submit" />  
</form>
```

Or, if it is a POST request that specifies a document, you need to use a multi-part form request:

```
...  
<form action="http://network_url/pg/cvapi/rest/xml" method="POST"  
enctype="multipart/form-data" >  
<input type="hidden" name="method" value="..." />  
<input type="hidden" name="auth_token" value="..." />  
<input type="file" name="file" />
```

More input fields

```
<input type="submit" />  
</form>
```

Response Format

The Clearvale API supports three response formats for data:

- Serialized PHP array
- XML
- JSON

In each of the three supported data formats there are two root elements:

- `status`. An integer type that specifies the method call status:
 - 0 indicates success
 - -1 indicates errors
 - positive numbers indicate method-specific failures
- `result`. An array type or `stdClass` that contains the result data when successful.

Using the API

The Clearvale API is categorized into groups based on functionality (see “Features Overview” on page 8 for a description of Clearvale features covered in these APIs). Each method name begins with a prefix that identifies the API to which it belongs. For example, `forums.create_topic` is part of the Forums API.

APIs	Description
Activity and Analytics	Get activity information from your network.
Blogs	Get lists of blogs, blog folders, and blog comments. Also, create and update blogs and so on.
Bookmarks	Gets bookmarks as well as creates new ones, and updates or deletes existing ones.
Calendar	Get calendar events as well as create new ones, and update or delete existing ones.
Communities	Get list of communities, add communities, remove members from communities, and so on.
CVAuth	Before calling a Clearvale API method that requires <code>auth_token</code> , use one of these methods to obtain a token, which has a lifetime of two hours.
Documents	Get list of documents for a category or community, create documents, delete documents, get document history and so on. “Documents” refer to files in Clearvale; videos, .jpg, .doc, .xls files, and so on can all be uploaded to Clearvale as files.
Forums	Create forums, delete forums, update forums, delete comments, and so forth. Use these methods for both community and network forums. NOTE: The <code>user_id</code> parameter refers to either a <code>user_guid</code> or a username that uniquely identifies a user.
Membership	Get list of members, add members, and update members. Also, includes APIS for working with personal and network groups.
Messages	Get information about Clearvale messages.
Microblog	Get information about microblogs.
Org	Create and mange organization structures for your network.
Page	Create, update, delete, and get wiki pages.
Poll	Create polls, delete polls, update polls, vote in polls, and so forth. Use these methods for community, network, or personal polls.
Profile	Update profile attributes, get user profile, and so on. NOTE: The <code>user_id</code> parameter refers to either a <code>user_guid</code> or a username that uniquely identifies a user.
Search	Search a network and return matching content.
Statistics	Get network statistics.
Task	Create tasks, delete tasks, update tasks, and so forth.

Activity

Get activity information from your network.

Method	Description
<code>activity.get_activities</code>	Get the activity stream for the current user.
<code>activity.get_user_activities</code>	Get the activities for one or all network members.
<code>activity.get_community_user_activities</code>	Get the community activities for one or more community members.
<code>activity.get_activities_on_content</code>	Get the activities related to content items.
<code>activity.get_activity_points_for_users</code>	Get the activity points for one, many, or all network members.
<code>activity.get_activity_points_for_content</code>	Get activity points for specific content items.
<code>activity.get_activity_points_for_communities</code>	Get activity points for communities.
<code>activity.get_followed_activities</code>	Get the Follow & Activity settings for a user.
<code>activity.set_followed_activities</code>	Set the Follow & Activity settings for a user.

The `activity.get_activities` API gets the activities that appear in the activity stream. The rest of these activity APIs take into account all activities that take place in the network, not just those that appear in the activity stream. For example, some activities such as rating content and tasks do not appear in the activity stream.

activity.get_activities

Get the activity stream for the current user.

Access

- Network members can get their activity stream.
- If you have a hybrid network, guests can get their activity stream.

Method

GET

Parameters

`options` (optional). Specify one or more of the following options:

- `locale`. Two character language identifier. The supported locale list is: `de`, `en`, `es`, `fr`, `it`, `ja`, `ko`, `nl`, `tw`, and `zh`. The default is English, `en`.
- `space_id`. The Clearvale space ID. Options are `0` for the network space or a community ID. The default is `0`.
- `limit`. The maximum number of activities to return. The default is `20`. In this case, the 20 most recent activities that appeared in your activity stream are returned.
- `comments`. The number of comments to return for the activities. Options are `-1` for all comments or a positive number for the number of comments to include. The default is `1`.
- `rid`. The river ID. Use this option to return activities that start from the river ID specified. You can “page” through the activity stream by using the river ID returned from a previous call to this method. The default value is null, which means to return the newest activities.
- `object_ids`. A list of object IDs, separated by commas. Use the objects IDs returned from the last API call for this method. The default value is null.
- `admin_mode`. Whether to operate in administrative mode or not. Options are `1` for administrative mode and `0` for non-administrative mode. You must be the network owner, supervisor, or moderator to use administrative mode.

Separate the options with the pipe (`|`). For example: `options=locale=fr|limit=50`

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the following in an array. Each record includes:

Element	Description
<code>actor_name</code>	Actor name.
<code>actor_guid</code>	Actor ID.
<code>action_name</code>	Action name, such as create.

Element	Description
template	<p>Activity template. The template looks as follows:</p> <pre>'%1\$s commented on %2\$s\'s %3\$s %4\$s'</pre> <p>Where:</p> <p>%1\$s is replaced with Member_Name %2\$s is replaced with Owning_Member %3\$s is replaced with content_item %4\$s is replaced with Content_name</p>
template_filler	<p>Activity template filler (an array of values to plug into the template).</p> <p>For example, template might look as follows:</p> <pre>"%1\$s shared %2\$s %3\$s with %4\$s",</pre> <p>And if template_filler looks like this:</p> <pre>"template_filler": ["Isabella Newton", "the file", "Clearvale - API List.doc", "Broadvision", ""],</pre> <p>It results in:</p> <pre>"Isabella Newton shared the file Clearvale - API List.do with Broadvision ",</pre>
space_name	Space name.
space_id	Space ID. 0 is network space. Other numbers are the community ID.
object_owner	Object owner email.
object_owner_id	Object owner ID.
object_type	Object type. Objects are based on the activity you conducted. If you commented on a blog, the object is the blog. If you joined a community, the object is the community.
object_name	Object name.
object_id	Object ID.
text_body	Text for the content item acted upon. Text is the content itself (such as a blog) or in the case of a file, the file description.
comments	An array of comments.
commentable	Does this activity type allow comments?
total_numbers	Total number of comments.
comments	Comments.
timestamp	Timestamp.
river_id	River ID.

activity.get_user_activities

Get the activities for one or all network members.

Access

- This method allows an anonymous caller to call without a token. If you call as an anonymous caller, you can only retrieve public activities.
- Network members can see the activities for other network participants. They will only see the content for which they have permission to view.
- If you have a hybrid network, guests can only use this API on users that they can see in the network. Also, if the user is unnamed to the guest, the user's name does not appear in the return results.

Method

GET

Parameters

`user_id` (required). The user ID or login (email address). Use 0 to return the activities for all members of the network.

`options` (optional). Specify one or more of the following options:

- `content_types`. One or more content type names separated by a comma. Possible content types are:

blog	forumtopic
bookmarks	microblog
dvault	page (wiki pages)
embedded	page_top (the top wiki page)
event_calendar	pages_welcome (the wiki welcome message)
file	poll
forum	task

The default is to use all content types.

- `limit`. The maximum number of matches to find. The default is 50.
- `sort_field`. The field on which to sort the results. The default is `activity_time`.
- `sort_direction`. The order in which to sort the results. Options are: `asc` for ascending or `desc` for descending.
- `from`. The beginning date of the date range. Use the format YYYYMMDD. For example, 20110407 is for April 7, 2011.
- `to`. The end date of the date range. Use the format YYYYMMDD.
- `admin_mode`. Whether to operate in administrative mode or not. Options are 1 for administrative mode and 0 for non-administrative mode. You must be the network owner, supervisor, or moderator to use administrative mode.

Separate the options with the pipe (`|`). For example:

```
options=content_types=blog,file|limit=100|from=20120301|to=20120630
```

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the following in an array. Each record includes:

actor_guid
actor_name
object_guid
object_name
object_type
object_owner_id
object_creation_time
object_modify_time
activity_name*
activity_time

*activity_name can be one of the following:

annotate (comments)	update	decline	sent feedback
create	view	add contact	sent invitation
delete	download	delete bookmark	sent message
follow	upload	join	sent recognition
negative rate	vote	post message board	view bookmark
positive rate	complete	replied to message	invite

activity.get_community_user_activities

Get the community activities for one or more community members.

Access

- This method allows an anonymous caller to call without a token. If you call as an anonymous caller, you can only retrieve public activities.
- Network members can only see the content for which they have permission.
- If you have a hybrid network, guests can only use this API on users that they can see in the network. Also, if the user is unnamed to the guest, the user's name does not appear in the return results.

Method

GET

Parameters

`community_id` (required). The community ID.

`user_id` (required). The user ID or login (email address) to limit the return to community activities of a particular community member. Use 0 to return the activities for all members of the community.

`options` (optional). Specify one or more of the following options:

- `content_types`. One or more content type names separated by a comma. Possible content types are:

blog	forumtopic
dvault	microblog
embedded	page (wiki pages)
event_calendar	page_top (the top wiki page)
file	pages_welcome (the welcome message)
forum	poll

The default is to use all content types.

- `limit`. The maximum number of matches to find. The default is 50.
- `sort_field`. The field on which to sort the results. The default is `activity_time`.
- `sort_direction`. The order in which to sort the results. Options are: `asc` for ascending or `desc` for descending.
- `from`. The beginning date of the date range. Use the format YYYYMMDD. For example, 20110407 is for April 7, 2011.
- `to`. The end date of the date range. Use the format YYYYMMDD.
- `admin_mode`. Whether to operate in administrative mode or not. Options are 1 for administrative mode and 0 for non-administrative mode. You must be the network owner, supervisor, or moderator to use administrative mode.

Separate the options with the pipe (|). For example:

```
options=content_types=blog,file|limit=100|from=20120301|to=20120630
```

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the following in an array. Each record includes:

actor_guid
actor_name
object_guid
object_name
object_type
object_owner_id
object_creation_time
object_modify_time
activity_name*
activity_time

*activity_name can be one of the following:

annotate (comments)	update	decline	sent feedback
create	view	add contact	sent invitation
delete	download	delete bookmark	sent message
follow	upload	join	sent recognition
negative rate	vote	post message board	view bookmark
positive rate	complete	replied to message	invite

activity.get_activities_on_content

Get the activities related to content items.

Access

- This method allows an anonymous caller to call without a token. If you call as an anonymous caller, you can only retrieve public activities.
- If you call with a token, you can only see the content for which you have permission. For example, if something is shared with the network, you will see it. If it is shared with a community of which you are not a member, you will not be able to retrieve those activities.
- If you have a hybrid network, guests can only use this API on content that they can view.

Method

GET

Parameters

`content_id` (required). The content ID. Use `0` to get all content items in the network.

`options` (optional). Specify one or more of the following options:

- `content_types`. One or more content type names separated by a comma. Possible content types are:

blog	forumtopic
bookmarks	microblog
dvault	page (wiki pages)
embedded	page_top (the top wiki page)
event_calendar	pages_welcome (the wiki welcome message)
file	poll
forum	task

- `limit`. The maximum number of matches to find. The default is `50`.
- `user_id`. The ID of the user who operated on the content. `0` indicates to user all users.
- `sort_field`. The field on which to sort the results. The default is `activity_time`.
- `sort_direction`. The order in which to sort the results. Options are: `asc` for ascending or `desc` for descending.
- `from`. The beginning date of the date range. Use the format `YYYYMMDD`. For example, `20110407` is for April 7, 2011.
- `to`. The end date of the date range. Use the format `YYYYMMDD`.
- `admin_mode`. Whether to operate in administrative mode or not. Options are `1` for administrative mode and `0` for non-administrative mode. You must be the network owner, supervisor, or moderator to use administrative mode.

Separate the options with the pipe (`|`). For example:

```
options=content_types=blog,file|limit=100|from=20120301|to=20120630
```

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the following in an array. Each record includes:

actor_guid	
actor_name	
object_guid	
object_name	
object_type	
object_owner_id	
object_creation_time	
object_modify_time	
activity_name*	Possible activities that can be conducted on content. See list below for possible activities.
activity_points	Points are negative when someone unfollows content.
activity_time	Time activity took place.

*activity_name can be one of the following:

annotate (comments)	update	decline	sent feedback
create	view	add contact	sent invitation
delete	download	delete bookmark	sent message
follow	upload	join	sent recognition
negative rate	vote	post message board	view bookmark
positive rate	complete	replied to message	invite

activity.get_activity_points_for_users

Get the activity points for one, many, or all network members.

Access

- Network members can see the activity points for other network participants.
- If you have a hybrid network, guests can only use this API on users that they can see in the network. Also, if the user is unnamed to the guest, the user's name does not appear in the return results.

Method

GET

Parameters

`user_ids` (required). One or more user IDs or logins (email addresses), separated by commas. Use 0 to return the activity points for all members.

`options` (optional). Specify one or more of the following options:

- `limit`. The maximum number of matches to find. The default is 50.
- `sort_field`. The field on which to sort the results. Use either `activity_points` or `user_name`. The default is `activity_points`.
- `sort_direction`. The order in which to sort the results. Options are: `asc` for ascending or `desc` for descending (default).
- `from`. The beginning date of the date range. Use the format YYYYMMDD. For example, 20110407 is for April 7, 2011.
- `to`. The end date of the date range. Use the format YYYYMMDD.
- `admin_mode`. Whether to operate in administrative mode or not. Options are 1 for administrative mode and 0 for non-administrative mode. You must be the network owner, supervisor, or moderator to use administrative mode.

Separate the options with the pipe (|). For example:

```
options=limit=100|sort_field=user_name|from=20120301|to=20120630
```

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the following in an array. Each record includes:

<code>user_guid</code>
<code>user_name</code>
<code>activity_points</code>

activity.get_activity_points_for_content

Get activity points for specific content items.

Access

- Network members can see the activity points for any content that they can view.
- If you have a hybrid network, guests can only use this API on content that they can view.

Method

GET

Parameters

`content_ids` (required). One or more content IDs, separated by commas. Use 0 to return the activity points for all content items.

`options` (optional). Specify one or more of the following options:

- `limit`. The maximum number of matches to find. The default is 50.
- `content_types`. One or more content type names separated by a comma. Possible content types are:

blog	forumtopic
bookmarks	microblog
dvault	page (wiki pages)
embedded	page_top (the top wiki page)
event_calendar	pages_welcome (the wiki welcome message)
file	poll
forum	task

The default is all content types.

- `sort_field`. The field on which to sort the results. The default is `activity_points`.
- `sort_direction`. The order in which to sort the results. Options are: `asc` for ascending or `desc` for descending (default).
- `from`. The beginning date of the date range. Use the format YYYYMMDD. For example, 20110407 is for April 7, 2011.
- `to`. The end date of the date range. Use the format YYYYMMDD.
- `admin_mode`. Whether to operate in administrative mode or not. Options are 1 for administrative mode and 0 for non-administrative mode. You must be the network owner, supervisor, or moderator to use administrative mode.

Separate the options with the pipe (|). For example:

```
options=limit=100|sort_field=user_name|from=20120301|to=20120630
```

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns On success, this method returns the following in an array. Each record includes:

<code>object_guid</code>
<code>object_name</code>
<code>object_name_owner</code>
<code>object_creation_time</code>
<code>object_modify_time</code>
<code>object_type</code>
<code>rating_positive</code>
<code>rating_negative</code>
<code>activity_points</code>

`activity.get_activity_points_for_communities`

Get activity points for communities.

Access

- Network members can see the activity points for all communities that they can view.
- If you have a hybrid network, guests can only use this API if they are a member of the community. Also, if any users are unnamed to the guest, the user's name does not appear in the return results.

Method

GET

Parameters

`community_ids` (required). One or more community IDs, separated by commas. Use 0 to return the activity points for all communities.

`options` (optional). Specify one or more of the following options:

- `limit`. The maximum number of matches to find. The default is 50.
- `sort_field`. The field on which to sort the results. The default is `activity_points`.
- `sort_direction`. The order in which to sort the results. Options are: `asc` for ascending or `desc` for descending (default).
- `from`. The beginning date of the date range. Use the format YYYYMMDD. For example, 20110407 is for April 7, 2011.
- `to`. The end date of the date range. Use the format YYYYMMDD.
- `admin_mode`. Whether to operate in administrative mode or not. Options are 1 for administrative mode and 0 for non-administrative mode. You must be the network owner, supervisor, or moderator to use administrative mode.

Separate the options with the pipe (`|`). For example:

```
options=limit=100|sort_field=community_name|from=20120301|to=20120630
```

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the following in an array. Each record includes:

community_guid
community_name
community_owner_id
community_creation_time
community_last_updated_time
community_member_number
activity_points

activity.get_followed_activities

Get the Follow & Activity settings for a user.

Access

- Network members can get the settings for themselves. Admins can get the settings for any user.

Method

GET

Parameters

`user_id` (required). The ID of the user's whose settings you want to get.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Invalid or expired token.
3	Invalid user ID.
4	Permission denied.

Returns

On success, this method returns a list of values corresponding to the settings of the Follow & Activity settings page. `0` indicates disabled, `1` indicates enabled.

writes_on_myactivitystream
on_mine
on_followed_content
message
task
complete_or_decline_task_created_by_me
update_task
shares
invites
add_content_community_follow
update_content_community_follow
instant_sendflag
daily_sendflag
weekly_sendflag
file
blog
forum
event
wiki
bookmark
poll
membership
profile

follow_on_network
follow_on_comment
follow_on_content_by_comments

activity.set_followed_activities

Set the Follow & Activity settings for a user.

Access

- Network members can change their own settings. Admins can change the settings for any user.

Method

SET

Parameters

`user_id` (required). The ID of the user.

`option` (required). A string with one or more of the following settings to set for the specified user. Use `0` to indicate disabled, use `1` to indicate enabled:

- `writes_on_myactivitystream`. When set to `1`, checks the **Writes on my page** setting for the user specified.
- `on_mine`. When set to `1`, checks the **Comments on or modifies my content** setting for the user specified.
- `on_followed_content`. When set to `1`, checks the **Comments on or modifies content I follow** setting for the user specified.
- `message`. When set to `1`, checks the **Sends me a message** setting for the user specified.
- `task`. When set to `1`, checks the **Assigns a task to me** setting for the user specified.
- `complete_or_decline_task_created_by_me`. When set to `1`, checks the **Accepts, Completes, or Declines a task that I assigned** setting for the user specified.
- `update_task`. When set to `1`, checks the **Updates or comments on a task assigned or owned by me** setting for the user specified.
- `shares`. When set to `1`, checks the **Shares content with me directly** setting for the user specified.
- `invites`. When set to `1`, checks the **Invites me to join a community** setting for the user specified.
- `add_content_community_follow`. When set to `1`, checks the **Adds new content to communities I follow** setting for the user specified.
- `update_content_community_follow`. When set to `1`, checks the **Updates content in communities I follow** setting for the user specified.
- `instant_sendflag`. When set to `1`, checks the **Send Email Notifications instantly** setting for the user specified.
- `daily_sendflag`. When set to `1`, checks the **Send a Daily Digest** setting for the user specified.
- `weekly_sendflag`. When set to `1`, checks the **Send a Weekly Digest** setting for the user specified.
- `file`. When set to `1`, checks the **File Activity** setting for the user specified.
- `blog`. When set to `1`, checks the **Blog Activity** setting for the user specified.

- `forum`. When set to `1`, checks the **Forum Activity** setting for the user specified.
- `event`. When set to `1`, checks the **Event Calendar Activity** setting for the user specified.
- `wiki`. When set to `1`, checks the **Wiki Activity** setting for the user specified.
- `bookmark`. When set to `1`, checks the **Bookmark Activity** setting for the user specified.
- `poll`. When set to `1`, checks the **Poll Activity** setting for the user specified.
- `membership`. When set to `1`, checks the **Participant Activity** setting for the user specified.
- `profile`. When set to `1`, checks the **Participant Profile Activity** setting for the user specified.
- `follow_on_network`. When set to `1`, checks the **Follow activity in the Network Space** setting for the user specified.
- `follow_on_comment`. When set to `1`, checks the **Automatically follow content items that I comment on** setting for the user specified.
- `follow_on_content_by_comments`. When set to `1`, checks the **Show comments made by participants I follow on unfollowed content** setting for the user specified.

Separate the options with the pipe (`|`). For example:

```
options='writes_on_myactivitystream=0|daily_sendflag=1'
```

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Invalid or expired token.
1	Invalid option.
3	Invalid user ID.
4	Permission denied.

Returns

On success, this method returns a list of values corresponding to the settings of the Follow & Activity settings page.

Analytics

In addition to the [Activity](#) APIs, you can use the `analytics.get_activity_points` API to get the activity points for a user. Note that this method calculates just the points for activities that appear in the activity stream.

Method	Description
<code>analytics.get_activity_points</code>	Get activity points for a given user.

NOTE: Previous versions of Clearvale supported additional analytic APIs. Those earlier APIs have been deprecated.

`analytics.get_activity_points`

Get activity points for a given user.

This API only includes points for activities that appear in the activity stream. Some activities do not appear in the stream, such as content ratings, microblogs, and some task activities. If you want a complete count of all activities, use `activity.get_activity_points_for_users` on [page 26](#).

Method `GET`

Parameters `user_id` (required). The user ID or login (email address).

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns On success, this method returns the number of activity points.

Request Example `http://network_url/pg/cvapi/rest/xml?method=analytics.get_activity_points&auth_token=df202eb08cfaf08f61c6f987b0d68e5e&user_id=36`

Blogs

Get lists of blogs, blog folders, and blog comments. Also, create and update blogs and so on.

Method	Description
blogs.get_folders_by_user	Get list of blog folders created by a specified user.
blogs.get_folders_for_space	Get list of blog folders for a specific community or the network space.
blogs.get_blogs_by_folder	Get list of blogs included in a specified folder.
blogs.get_blogs_by_user	Get the list of blogs created by a user.
blogs.get_blogs_by_community	Get the list of blog posts for a community.
blogs.get_comments_by_blog	Get the list of comments posted to a given blog.
blogs.create_blog	Create a blog in your personal space.
blogs.update_blog	Update a blog.
blogs.delete_blog	Delete a blog.
blogs.add_comment	Add a comment to a blog.
blogs.delete_comment	Delete a comment.
blogs.create_category	Create a blog folder.
blogs.update_category	Update a blog folder.
blogs.delete_category	Delete a blog folder.
blogs.get_blog_by_id	Get a blog by blog ID.
blogs.share_blog	Share a blog.
blogs.unshare_blog	Unshare a blog.

blogs.get_folders_by_user

Get list of blog folders created by a specified user.

Access

- Network members can get the blog folders of different participants in the network.
- If you have a hybrid network, guests can only retrieve the blog folders in their personal space.

Method

GET

Parameters

`user_id` (required). The user ID or login (email address). 0 indicates caller.

`limit` (optional). Specifies maximum number of folders to return. When not specified or set to 0, all are folders returned.

`offset` (optional). 0 is default.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the following in an array. Each record includes:

id
icon
title
description
tags
privacy
time_created
owner

Example

`http://network_url/pg/cvapi/rest/xml?method=blogs.get_folders_by_user&auth_token=7346c04260223a7870f122b835158300&user_id=3&limit=100`

blogs.get_folders_for_space

Get list of blog folders for a specific community or the network space.

Access

- Network members can get the blog folders of communities to which they belong.
- If you have a hybrid network, guests can use this API to get the blog folders of guest or external communities to which they belong. If they have access to the network page, they can get the network folders.

Method

GET

Parameters

`space_id` (required). The community ID, or 0 to indicate the network space.

`limit` (optional). Specifies maximum number of folders to return. When not specified or set to 0, all folders are returned.

`offset` (optional). Default is 0.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the following in an array. Each record includes:

id
icon
title
description
tags
privacy
time_created
owner

blogs.get_blogs_by_folder

Get list of blogs included in a specified folder.

Access

- Network members can get the blogs within a folder, assuming they have permissions to see the folder.
- If you have a hybrid network, guests can get a list of all blogs in a specific folder, assuming they have permission to see the folder.

Method

GET

Parameters

`cat_id` (required). Category ID.

`limit` (optional). Specifies the maximum number of blogs to return. When not specified or set to 0, it returns all blogs.

`offset` (optional). Default is 0.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the following in an array. Each record includes:

id
title
body
tags
privacy
folder
show_home_page
allow_comments
time_created
owner_guid
owner
positive_votes
negative_votes
comments_total

Example

`http://network_url/pg/cvapi/rest/xml?method=blogs.get_blogs_by_folder&auth_token=7346c04260223a7870f122b835158300&cat_id=3&limit=100`

blogs.get_blogs_by_user

Get the list of blogs created by a user.

Access

- If you have a hybrid network, guests can get a list of all blogs by a specific user assuming they can see the user. Also, if the user is unnamed to the guest, then user's name does not appear in the return results.

Method

GET

Parameters

`user_id` (required). The user ID or login (email address). 0 indicates the caller.

`limit` (optional). Specifies the maximum number of blogs to return. When not specified or set to 0, it returns all blogs.

`offset` (optional). Default is 0.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the following in an array. Each record includes:

id
title
body
tags
privacy
folder
show_home_page
allow_comments
time_created
owner

Example

`http://network_url/pg/cvapi/rest/xml?method=blogs.get_blogs_by_user&auth_token=7346c04260223a7870f122b835158300&user_id=3&limit=100`

blogs.get_blogs_by_community

Get the list of blog posts for a community.

Access

- If you have a hybrid network, guests can get a list of all blog posts for communities of which they are a member.

Method

GET

Parameters

`community_id` (required). The community ID.

`limit` (optional). Specifies the maximum number of blog posts to return. When not specified or set to 0, it returns all blog posts.

`offset` (optional). Default is 0.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to an expired token.
7	Empty or invalid community ID.
6	Permission denied.

Returns

On success, this method returns the following in an array. Each record includes:

id
title
body
tags
privacy
folder
show_home_page
allow_comments
time_created
owner_guid
owner
positive_votes
negative_votes
comments_total

Example

`http://network_url/pg/cvapi/rest/xml?method=blogs.get_blogs_by_community&auth_token=7346c04260223a7870f122b835158300&user_id=31&limit=10&offset=20`

blogs.get_comments_by_blog

Get the list of comments posted to a given blog.

Access

- This method allows an anonymous caller to call without a token. If you call as an anonymous caller, you can only retrieve comments on public blogs.
- Network members can only see the comments on blogs for which they have permission.
- If you have a hybrid network, guests can use this API only if the blog is shared with the guest or it is their own blog.

Method

GET

Parameters

`blog_id` (required). Blog ID.

`limit` (optional). Specifies the maximum number of blog comments to return. When not specified or set to 0, it returns all comments.

`offset` (optional). Default is 0.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the following in an array. Each record includes:

<code>id</code>	Comment ID.
<code>desc</code>	Comment.
<code>time_created</code>	Time that the comment was made.
<code>owner</code>	Person who added the comment. The value <code>unknown user</code> is returned if the blog is shared across networks in an ecosystem and the person who made the comment is not a member on a particular network.
<code>owner_icon</code>	Avatar for the person adding a comment. If <code>unknown user</code> is returned for <code>owner</code> , the <code>owner_icon</code> is an empty string.

Example

`http://network_url/pg/cvapi/rest/xml?method=blogs.get_comments_by_blog&auth_token=7aabb5470e6fadcc610e30c0211d1ccd&blog_id=31&limit=100`

blogs.create_blog

Create a blog in your personal space.

To share the blog with the network space or a community space, use [blogs.share_blog](#) after creating the blog.

Access

- Network members can use this API to create a blog.
- If you have a hybrid network, guests can use this API to create a blog. Where they can post the blog is limited by their network permissions. That is, if they have access to the network page, they can share the blog with the network or general public.

Method

POST

Parameters

`title` (required). Blog title.

`body` (required). Blog body.

`options` (optional). Specify one or more of the following options:

- `tags`. Blog tags. Separate each tag by comma.
- `folders`. Blog folder IDs. Separate each folder ID with comma. You can only specify blog folders in your personal space.
- `show_home_page`. Possible values are:
 - `0` indicates not to show on the Network page.
 - `1` indicates to show on the Network page (default value).
 - `2` indicates to use the default value for the network.

If your network is configured so that only administrators or moderators can write to the Network page, then you must have one of these roles to use the `show_home_page` setting.

- `allow_comments`. `0` indicates allow comments, `1` indicates no.
- `use_html`. One of the following:
 - `yes` indicates that HTML is allowed in the blog body. This is the default behavior.
 - `no` indicates that HTML is not allowed in the blog body.
- `share_with_groups`. Indicates the community IDs of any communities with which to share the blog. Separate the community IDs with commas.
- `share_with_network`. One of the following:
 - `1` indicates to share the blog with the network.
 - `0` indicates not to share the blog with the network.
- `general_public`. One of the following:
 - `1` indicates to let the general public see the blog.
 - `0` indicates not to show the blog to the general public.

Separate the options with the pipe (`|`). For example:

```
options=show_home_page=1|allow_comments=1
```

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the ID for the new blog

blogs.update_blog

Update a blog.

Access

- If you have a hybrid network, guests can use this API to update a blog to which they have access. Where they can post the blog is limited by their network permissions. That is, if they have access to the network page, they can share the blog with the network or general public.

Method

POST

Parameters

`blog_id` (required). Blog ID.

`options` (optional). Specify one or more of the following options:

- `title`. Blog title.
- `body`. Blog body.
- `tags`. Blog tags. Separate each tag with a comma.
- `folders`. Blog folder ID. Separate each folder id by comma.
- `show_home_page`. Possible values are:
 - `0` indicates not to show on the Network page.
 - `1` indicates to show on the Network page (default value).
 - `2` indicates to use the default value for the network.

If your network is configured so that only administrators or moderators can write to the Network page, then you must have one of these roles to use the `show_home_page` setting.

- `allow_comments`. `0` indicates allow comments, `1` indicates no.
- `use_html`. One of the following:
 - `yes` indicates that HTML is allowed in the blog body. This is the default behavior.
 - `no` indicates that HTML is not allowed in the blog body.
- `share_with_groups`. Indicates the groups IDs of any groups with which to share the blog. Separate the community IDs with commas.
- `share_with_network`. One of the following:
 - `1` indicates to share the blog with the network.
 - `0` indicates not to share the blog with the network.
- `general_public`. One of the following:
 - `1` indicates to let the general public see the blog.
 - `0` indicates not to show the blog to the general public.

Separate the options with the pipe (`|`). For example:

```
options=show_home_page=1|allow_comments=1
```

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns On success, this method returns the blog ID.

blogs.delete_blog

Delete a blog.

Access

- If you have a hybrid network, guests can use this API to delete any blog they have permission to edit.

Method GET

Parameters `blog_id` (required). Blog ID.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns On success, this method returns `1`.

blogs.add_comment

Add a comment to a blog.

Access

- If you have a hybrid network, guests can use this API to comment on blogs that they permission to view.

Method

POST

Parameters

`blog_id` (required). Blog ID.

`comment` (required). Comment.

`options` (optional). Specify the following option:

- `use_html`. One of the following:
 - `yes` indicates that HTML is allowed in the blog body. This is the default behavior.
 - `no` indicates that HTML is not allowed in the blog body.

For example: `options=use_html=no`

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the comment ID.

blogs.delete_comment

Delete a comment.

Access

- If you have a hybrid network, guests can use this API to delete any comment they added or that they have permission to edit.

Method

GET

Parameters

`blog_id` (required). Blog ID.

`comment_id` (required). Comment ID.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
8	Failed due to invalid comment ID.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns `1`.

blogs.create_category

Create a blog folder.

Access

- If you have a hybrid network, guests can use this API to create a blog folder in their personal space.

Method

POST

Parameters

`title` (required). Folder title.

`options` (optional). Specify one or more of the following options:

- `icon`. Folder icon.
- `description`. Folder description.
- `tags`. Folder tags; separate each with a comma.

Separate the options with the pipe (`|`). For example: `options=description=product reviews|tags=reviews`

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns category ID.

blogs.update_category

Update a blog folder.

Access

- If you have a hybrid network, guests can use this API to update a blog folder in their personal space.

Method

POST

Parameters

`cat_id` (required). Folder ID.

`options` (optional). Specify one or more of the following options:

- `icon`. Folder icon.
- `title`. Folder title.
- `description`. Folder description.
- `tags`. Folder tags; separate each tag with a comma.

Separate the options with the pipe (`|`). For example: `options=description=product reviews|tags=reviews`

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns category ID.

blogs.delete_category

Delete a blog folder.

Access

- If you have a hybrid network, guests can use this API to delete a blog folder in their personal space.

Method

GET

Parameters

`cat_id` (required). Folder ID.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns `1`.

blogs.get_blog_by_id

Get a blog by blog ID.

Access

- If you have a hybrid network, guests can use this API to get a blog assuming that they can view the blog.

Method

GET

Parameters

`blog_id` (required). Blog ID.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the following in an array. Each record includes:

<code>id</code>
<code>title</code>
<code>desc</code>
<code>count_comments</code>
<code>tags</code>
<code>owner</code>
<code>owner_icon</code>
<code>time_created</code>

Example

`http://network_url/pg/cvapi/rest/xml?method=blogs.get_blog_by_id&auth_token=df202eb08cfaf08f61c6f987b0d68e5e&blog_id=1234`

blogs.share_blog

Share a blog.

Access

- Network members can share a blog with the network or with specific communities to which they belong.
- If you have a hybrid network, guests can use this API to share a blog to guest and external communities to which they belong. Additionally, if they have access to the network page, they can share the blog with the network space.

Method

POST

Parameters

`blog_id` (required). Blog ID.

`space_ids` (required). Specify the space IDs with which to share the blog. Separate the space IDs with commas.

- `0` means network space.
- Community ID means a particular community.
- Network ID means a particular network if you have an ecosystem of networks.

`options` (optional). Specify the following option:

- `folder_ids`. Blog folder IDs from the network or community spaces. Separate multiple blog folder IDs with commas.

For example: `options=folder_ids=1234,1334`

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the string “success.”

blogs.unshare_blog

Unshare a blog.

Access

- Network members can use this API to unshare a blog.
- If you have a hybrid network, guests can use this API to unshare a blog.

Method

POST

Parameters

`blog_id` (required). Blog ID.

`space_ids` (required). Specify the space IDs with which to unshare the blog.

- 0 means network space.
- Community ID means a particular community.

Separate the space IDs with commas.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the string “success.”

Bookmarks

Gets bookmarks as well as creates new ones, and updates or deletes existing ones.

Method	Description
<code>bookmarks.create_bookmark</code>	Creates a bookmark.
<code>bookmarks.update_bookmark</code>	Updates a bookmark.
<code>bookmarks.delete_bookmark</code>	Deletes a bookmark.
<code>bookmarks.get_bookmarks</code>	Gets a list of bookmarks.

`bookmarks.create_bookmark`

Creates a bookmark.

Access

- Network members can create bookmarks.
- If you have a hybrid network, guests can use this API to add a bookmark on their personal page. Additionally, if they have access to the network page, they can share the bookmark to the network space.

Method

POST

Parameters

`title` (required). A title for the bookmark.

`address` (required). The URL for the bookmark.

`privacy` (required). Determines who can see the bookmark. Use `0` for Only Me, `1` for network members, and `2` for General Public.

`options` (optional). Specify one or more of the following options:

- `description`. An optional bookmark description.
- `tags`. One or more keywords for the bookmark. Separate keywords with commas.
- `use_html`. One of the following:
 - `yes` indicates that HTML is allowed in the bookmark description. This is the default behavior.
 - `no` indicates that HTML is not allowed in the bookmark description.

Separate the options with the pipe (`|`). For example: `options=description=Shortcut to annual report|tags=report`

Chapter 2 APIs

Using the API

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the bookmark ID.

bookmarks.update_bookmark

Updates a bookmark.

Access

- Network members can update their own bookmarks.
- If you have a hybrid network, guests can use this API to update their own bookmarks.

Method

POST

Parameters

`link_id` (required). ID of the bookmark to update.

`options` (optional). Specify one or more of the following options:

- `title`. A title for the bookmark.
- `address`. The URL for the bookmark.
- `privacy`. Determines who can see the bookmark. Use `0` for Only Me, `1` for network members, and `2` for General Public.
- `tags`. One or more keywords for the bookmark. Separate keywords with commas.
- `description`. A bookmark description.
- `use_html`. One of the following:
 - `yes` indicates that HTML is allowed in the bookmark description. This is the default behavior.
 - `no` indicates that HTML is not allowed in the bookmark description.

Separate the options with the pipe (`|`). For example: `options=privacy=1|tags=report`

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns `1`.

bookmarks.delete_bookmark

Deletes a bookmark.

Access

- Network members can delete their own bookmarks.
- If you have a hybrid network, guests can use this API to delete their own bookmarks.

Method

POST

Parameters

`link_id` (required). ID of the bookmark to update.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns `1`.

bookmarks.get_bookmarks

Gets a list of bookmarks.

Access

- This method allows an anonymous caller to call without a token. If you call as an anonymous caller, you can only retrieve public bookmarks.
- Network members can only see the content for which they have permission. For example, if something is shared with the network, they will see it.
- If you have a hybrid network, guests can use this API to get their bookmarks. Additionally, if they have access to the network page, they can get the bookmarks that participants shared to the network space.

Method

GET

Parameters

`user_id` (required). The user ID or login (email address) of the member whose bookmarks you want to retrieve. If `0`, then gets bookmarks for all users.

`options` (optional). Specify one or more of the following options:

- `limit`. Specifies the maximum number of matches to return. If not specified or set to `0`, `50` is used.
- `offset`. The offset from the first match. If not specified, `0` is used.

Separate the options with the pipe (`|`). For example: `options=limit=10|offset=0`

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns an array of bookmarks with the following attributes:

guid	Bookmark ID.
subtype	The value 21 to indicate that this is a bookmark.
owner_guid	Bookmark owner ID.
container_guid	Bookmark owner ID.
site_guid	Site ID.
access_id	Who can access the bookmark: <ul style="list-style-type: none">• 0 indicates a private bookmark and only you can see it.• 1 indicates a network-level bookmark.• 2 indicates a public bookmark.
time_created	Time that the bookmark was created.
time_updated	Time that the bookmark was updated.
title	Title of the bookmark.
address	URL of the bookmark.
description	Detailed information about the bookmark.

Calendar

Gets calendar events as well as creates new ones, and updates or deletes existing ones.

Method	Description
<code>calendar.get_events</code>	Gets a list of events within a date range.
<code>calendar.create_event</code>	Creates a new event.
<code>calendar.update_event</code>	Updates an existing event.
<code>calendar.delete_event</code>	Deletes an event.
<code>calendar.set_icon</code>	Sets an icon for an event.

`calendar.get_events`

Gets a list of events within a date range.

This method relies on Lucene for retrieving the data. If an event is created and this method is called before the indexer has run, then this method will not return the event.

Access

- This method allows an anonymous caller to call without a token. If you call as an anonymous caller, you can only retrieve public events.
- If you call with a token, you can only see the events for which you have permission. For example, if something is shared with the network, you will see it. If it is shared with a community of which you are not a member, you will not be able to retrieve those events.
- If you have a hybrid network, guests can use this API to get events from their personal space or communities to which they belong. Additionally, if they have access to the network page, they can get events from the network space.

Method

GET

Parameters

`timezone_offset` (required). The span between your local timezone and GMT0 timezone. Use the format `H` or `HH:MM` for the desired offset. For example:

- If your local timezone is GMT- 8, then `timezone_offset` is `-8`.
- If your local timezone is GMT+8, then `timezone_offset` is `8`.
- If your local timezone is GMT- 5:30, then `timezone_offset` is `-5:30`.

`from` (required). The beginning date of the date range. Use the format YYYYMMDD. For example, 20110407 is for April 7, 2011.

`to` (required). The end date of the date range. Use the format YYYYMMDD.

`community_id` (required). Communities for which to find events. Separate multiple community IDs with a colon `:`. Use `0` for all communities. If you use `0` for `community_id`, you cannot use `0` for `user_id`. Instead, specify the specific user IDs.

`user_id` (required). The users who created the events. Separate multiple user IDs or logins (email addresses) with a colon ':'. Use 0 to get the events created by all users. If you use 0 for `user_id`, you cannot use 0 for `community_id` as well. Instead, specify the specific community IDs.

`options` (optional). Specify one or more of the following options:

- `limit`. Specifies the maximum number of matches to return. If not specified or set to 0, 50 is used.
- `offset`. The offset from the first match. If not specified, 0 is used.
- `sort`. How to sort the search results. Possible settings are:
 - `time_updated`. The time the event was last updated.
 - `time_created`. The time the event was created.
 - `name`. The name of the event. (Default)
 - `creator_name`. The name of the person who created the event.
 - `owner_name`. The owner of the event.

Precede the keyword with + to indicate an ascending sort order or - to indicate a descending sort order.

Separate the options with the pipe (|). For example: `options=limit=10|sort=time_updated`

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the following in an array. Each record includes:

<code>guid</code>	Event ID.
<code>subtype</code>	The value 11, which indicates that this is an event.
<code>owner_guid</code>	User ID of the person who created the event.
<code>container_guid</code>	Can be one of the following: <ul style="list-style-type: none">• Community ID if the event belongs to a community• Owner ID if it is a personal event.
<code>site-guid</code>	Site ID.
<code>access-id</code>	Who can access the event: <ul style="list-style-type: none">• 0 indicates a private event and only you can see it.• 1 indicates a network-level event.• 2 indicates a public event.
<code>title</code>	Event title.
<code>location</code>	Event location.
<code>searchable_title</code>	Title of the event for searches. Same as the event title.

description	You can ignore this.
long_description	Details of the event.
time_created	Time that the event was created.
time_updated	Time that the event was updated.
start_date	Date that the event starts.
end_date	Date that the event ends.
start_time	Time that the event starts, for example: 11:30 AM
end_time	Time that the event ends, for example: 2:00 PM
tags	A comma-separated list of tags relevant to this event.
fees	Event fees.
contact	Event contact.
importance	Importance setting of the event. Either normal or important.
author_guid	Empty unless the event is at the community level. If the event is created in a community, this indicates the event owner ID.
event_type	The event type. Possible value: <code>event</code> .
site_selected	Whether or not to display the event on the Network page: <ul style="list-style-type: none">• <code>1</code> means show on Network page.• <code>2</code> means do not show on Network page.
has_icon	Whether the event icon been set or not. <code>1</code> indicates that an icon is set. <code>0</code> indicates there is no icon.

calendar.create_event

Creates a new event.

Access

- Network members can create new events.
- If you have a hybrid network, guests can use this API to create events in their personal space or in communities to which they belong.

Method

POST

Parameters

`title` (required). A title for the event.

`timezone_offset` (required). The span between your local timezone and GMT0 timezone. Use the format `H` or `HH:MM` for the desired offset. For example:

- If your local timezone is GMT- 8, then `timezone_offset` is `-8`.
- If your local timezone is GMT+8, then `timezone_offset` is `8`.
- If your local timezone is GMT- 5:30, then `timezone_offset` is `-5:30`.

The timezone offset applies to the starting date, ending date, starting time, and ending time specified for the event.

`options` (optional). Specify one or more of the following options:

- `start_date`. The date when the event will start. Use the format YYYYMMDD. For example, `20110407` is for April 7, 2011.
- `end_date`. The date when the event will end. Use the format YYYYMMDD.
- `location`. Where the event will take place.
- `start_time`. The `start_time` should be in 24 hour clock and in format HHMM. MM must be `00`, `15`, `30` or `45`. If there is more than one tag, they should be separated by a comma.
- `end_time`. The `end_time` should be in 24 hour clock and in format HHMM. MM must be `00`, `15`, `30` or `45`. If there is more than one tag, they should be separated by a comma.
- `fees`. Any event fees.
- `contact`. The event contact.
- `tags`. Tags for the event. If more than one tag, separate with commas.
- `details`. The event details.
- `privacy`. The privacy for the event. Use `0` for Only Me, `1` for network members, and `2` for General Public.
- `importance`. The importance of this event. Use `0` for normal or `1` for important. `0` is the default.
- `show_home_page`. Whether to show the event on the Network page or not. Use `0` for Yes or `1` for No.

If your network is configured so that only administrators or moderators can write to the Network page, then you must have one of these roles to use the `show_home_page` setting.

- `community_id`. The community in which to create the event. Use `0` for no community; the event is created in the user's space.
- `use_html`. One of the following:
 - `yes` indicates that HTML is allowed in the event details. This is the default behavior.

- `no` indicates that HTML is not allowed in the event details.

Separate the options with the pipe (`|`). For example: `options=location=New York City|start_time=1000|end_time=1100`

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the new event ID.

calendar.update_event

Updates an existing event.

Access

- Network members can update their own events.
- If you have a hybrid network, guests can only update their own events.

Method

POST

Parameters

`event_id` (required). Event ID.

`timezone_offset`. The span between your local timezone and GMT0 timezone. Use the format `H` or `HH:MM` for the desired offset. For example:

- If your local timezone is GMT- 8, then `timezone_offset` is `-8`.
- If your local timezone is GMT+8, then `timezone_offset` is `8`.
- If your local timezone is GMT- 5:30, then `timezone_offset` is `-5:30`.

The timezone offset applies to the starting date, ending date, starting time, and ending time specified for the event.

`options` (optional). Specify one or more of the following options:

- `title`. The title of the event.
- `start_date`. The date when the event will start. Use the format YYYYMMDD. For example, `20110407` is for April 7, 2011.
- `end_date`. The date when the event will end. Use the format YYYYMMDD.
- `location`. Where the event will take place.
- `start_time`. The `start_time` should be in 24 hour clock and in format HHMM. MM must be `00`, `15`, `30` or `45`. If there is more than one tag, they should be separated by a comma.
- `end_time`. The `end_time` should be in 24 hour clock and in format HHMM. MM must be `00`, `15`, `30` or `45`. If there is more than one tag, they should be separated by a comma.
- `fees`. Any event fees.
- `contact`. The event contact.
- `tags`. Tags for the event. If more than one tag, separate with commas.
- `details`. The event details.
- `privacy`. The privacy for the event. Use `0` for Only Me, `1` for network members, and `2` for General Public.
- `importance`. The importance of this event. Use `0` for normal or `1` for important. `0` is the default.
- `show_home_page`. Whether to show the event on the Network page. Use `0` for Yes or `1` for No. If your network is configured so that only administrators or moderators can write to the Network page, then you must have one of these roles to use the `show_home_page` setting.
- `use_html`. One of the following:
 - `yes` indicates that HTML is allowed in the event details. This is the default behavior.
 - `no` indicates that HTML is not allowed in the event details.

Separate the options with the pipe (`|`). For example:

```
options=location=headquarters|start_time=1000|end_time=1100
```

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns `true`.

calendar.delete_event

Deletes an event.

Access

- Network members can delete their own events.
- If you have a hybrid network, guests can only delete their own events.

Method

GET

Parameters

`event_id` (required). Event ID.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns `true`.

calendar.set_icon

Sets an icon for an event.

This method must be called through a multi-part/form data POST request (see RFC1867 for more details as well as “Request Format” on page 14) that must include the parameters listed below and the icon file to be uploaded.

Access

- Network members can set icons for their own events.
- If you have a hybrid network, guests can only set icons for their own events.

Method

POST (multi-part/form data)

Parameters

`event_id` (required). Event ID.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns `true`.

Communities

Get list of communities, add communities, remove members from communities, and so on.

Method	Description
<code>communities.get_communities_for_user</code>	Get the list of network communities for a user.
<code>communities.add_user_to_community</code>	Add a user to a community.
<code>communities.remove_user_from_community</code>	Remove a user from a community.
<code>communities.add_community</code>	Add a new community.
<code>communities.get_community_by_id</code>	Get a community based on its community ID.
<code>communities.get_community_by_name</code>	Get a community by its name.
<code>communities.get_all_communities</code>	Get all communities of which you are a member.
<code>communities.get_communities_by_folder</code>	Get all the communities in a community folder.
<code>communities.get_folders</code>	Get all community folders that exist in the network.

communities.get_communities_for_user

Get the list of network communities for a user.

Access

- Network members can get a list of communities for network participants.
- If you have a hybrid network, guests can only get the list of communities for user's they can see in the network.

Method

GET

Parameters

`user_id` (required). The user ID or the login (email address).

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the following in an array. Each record includes:

<code>id</code>	Community ID.
<code>name</code>	Community name.
<code>owner_id</code>	Community owner ID.
<code>owner</code>	If the community's owner ID is equal to the user's ID, this field is set to Yes. If not, this field is omitted.
<code>time_created</code>	Time community was created.
<code>time_modified</code>	Time community was last modified.

Example

```
http://network_url/pg/cvapi/rest/xml?method=communities.get_communities_for_user&auth_token=df202eb08cfaf08f61c6f987b0d68e5e&user_id=john.doe@broadvision.com
```

communities.add_user_to_community

Add a user to a community.

Access

- You must be a network administrator or the community administrator to add a member to a community.
- If you have a hybrid network, guests can only add users to guest or external communities of which they are a community administrator.

Method

POST

Parameters

`user_id` (required). The user ID or login (email address).

`community_id` (required). Community ID of the community that the user should become a member of.

`options` (optional). Specify the following options:

- `admin_mode`. Whether to operate in administrative mode or not. Options are `1` for administrative mode and `0` for non-administrative mode. You must be the network owner, supervisor, or moderator to use administrative mode.

Status Code

Code	Description
3	User is already a member of the community.
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns a success message.

communities.remove_user_from_community

Remove a user from a community.

Access

- You must be a network administrator or the community administrator to remove people other than yourself from a community.
- If you have a hybrid network, guests can only remove participants from guest or external communities of which they are a community administrator.

Method

POST

Parameters

`user_id` (required). The user ID or login (email address).

`community_id` (required). Community ID of the community from which to remove the user.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns a success message.

communities.add_community

Add a new community.

Access

- You just have permission to create communities within the network.
- If you have a hybrid network, guests cannot use this API.

Method

POST

Parameters

`community_name` (required). The name of the community.

`options` (optional). Specify one or more of the following options:

- `contact_person`. The person to contact regarding the community.
- `contact_phone`. The phone number of the community contact.
- `description`. The description of the community.
- `contact_email`. The email address of the community contact.
- `show_on_network_page`. Whether to show the community on the Network page or not. Use `0` for Yes or `1` for No.
- `membership`. Whether the community is open to anyone or requires an invitation. Options are:
 - `open`
 - `invitation_only`

The default is `invitation_only`.

- `space_type`. The type of community to create. For regular networks, use `internal`. For hybrid networks, the options are: `internal`, `guest`, or `external`.
The default is `internal`.
- `privacy`. The privacy for the community. Use `1` for network members, `2` for General Public, and `3` for community members. The default is `network_members`.
- `use_html`. One of the following:
 - `yes` indicates that HTML is allowed in the community description. This is the default behavior.
 - `no` indicates that HTML is not allowed in the community description.

Separate the options with the pipe (`|`). For example: `options=contact_person=Jeremy Hull|contact_phone=408-727-1000|show_on_network_page=0`

If you have a hybrid network, and set `space_type` to `guest`, `membership` must be `invitation_only` and `privacy` must be `3`.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the ID of the new community.

communities.get_community_by_id

Get a community based on its community ID.

Access

- This method allows an anonymous caller to call without a token. If you call as an anonymous caller, you can only retrieve communities that are open to the general public.
- Network members can retrieve communities of which they are a member or with the privacy set to network.
- If you have a hybrid network, guests can only get guest communities of which they are member or external communities.

Method

GET

Parameters

`community_id` (required). The ID of the community to get.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the following in an array. Each record includes:

- `community_guid`
- `community_name`
- `community_description`
- `community_contact_person`
- `community_contact_phone`
- `community_contact_email`
- `community_membership`
- `space_type`
- `community_privacy`
- `community_show_on_network_page`
- `community_creation_time`
- `community_modify_time`
- `community_owner_id`
- `community_owner_name`

communities.get_community_by_name

Get a community by its name.

Access

- This method allows an anonymous caller to call without a token. If you call as an anonymous caller, you can only retrieve communities that are open to the general public.
- Network members can retrieve communities of which they are a member or with the privacy set to network.
- If you have a hybrid network, guests can only get guest communities of which they are member or external communities.

Method

GET

Parameters

`community_name` (required). The name of the community.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the following in an array. Each record includes:

<code>community_guid</code>
<code>community_name</code>
<code>community_description</code>
<code>community_contact_person</code>
<code>community_contact_phone</code>
<code>community_contact_email</code>
<code>community_membership</code>
<code>space_type</code>
<code>community_privacy</code>
<code>community_show_on_network_page</code>
<code>community_creation_time</code>
<code>community_modify_time</code>
<code>community_owner_id</code>
<code>community_owner_name</code>

communities.get_all_communities

Get all communities of which you are a member.

Access

- Network members can retrieve communities of which they are a member.
- If you have a hybrid network, guests get only communities to which they belong.

Method

GET

Parameters

`options` (optional). Specify one or more of the following options:

- `limit`. The maximum number of communities to return.
- `offset`. The offset from the first match. If not specified, `0` is used.
- `admin_mode`. Whether to operate in administrative mode or not. If you have the network owner, supervisor, or moderator role, use administrative mode to get all communities, even communities of which you are not a member. Options are `1` for administrative mode and `0` for non-administrative mode.

Separate the options with the pipe (`|`). For example: `options=limit=10|offset=10`

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the following in an array. Each record includes:

<code>community_guid</code>	
<code>community_name</code>	
<code>community_creation_time</code>	
<code>community_modify_time</code>	
<code>community_owner_id</code>	
<code>community_owner_name</code>	
<code>community_icon</code>	
<code>community_privacy</code>	Returns General Public, Community Members, or Network Members
<code>space_type</code>	Returns internal, guest, or external
<code>community_membership</code>	Returns open or invitation_only

communities.get_communities_by_folder

Get all the communities in a community folder.

Access

- This method allows an anonymous caller to call without a token. If you call as an anonymous caller, you can only retrieve communities that are open to the general public.
- Network members can retrieve communities of which they are a member or with the privacy set to network or public.
- If you have a hybrid network, guests can only get guest communities of which they are member or external communities.

Method

GET

Parameters

`folder_id` (required). The community folder ID.

`options` (optional). Specify one or more of the following options:

- `limit`. The maximum number of communities to return.
- `offset`. The offset from the first match. If not specified, 0 is used.
- `admin_mode`. Whether to operate in administrative mode or not. If you have the network owner, supervisor, or moderator role, use administrative mode to get all communities, even communities of which you are not a member. Options are 1 for administrative mode and 0 for non-administrative mode.

Separate the options with the pipe (|). For example: `options=limit=10|offset=10`

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the following in an array. Each record includes:

<code>community_guid</code>	
<code>community_name</code>	
<code>community_creation_time</code>	
<code>community_modify_time</code>	
<code>community_owner_id</code>	
<code>community_owner_name</code>	
<code>community_icon</code>	
<code>community_privacy</code>	Returns General Public, Community Members, or Network Members
<code>space_type</code>	Returns internal, guest, or external
<code>community_membership</code>	Returns open or invitation_only

communities.get_folders

Get all community folders that exist in the network.

Access

- Community folders have no privacy, so anyone can get all folders in the network.
- For public networks, this method allows an anonymous caller to call without a token. They also can retrieve all community folders.

Method

GET

Parameters

None.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the following in an array. Each record includes:

folder_id	Community folder ID
folder_name	Community folder name
time_created	Date and time community folder was created.

CVAuth

Most Clearvale API methods require that you provide an authentication token. Use the `cvauth.authenticate` or `cvauth.get_token` method to obtain a token. Pass the token to the Clearvale API methods by using the `auth_token` parameter. (For more information on using the `auth_token` parameter, see “Request Format” on page 14.) The token has a lifetime of two hours.

Method	Description
<code>cvauth.authenticate</code>	Authenticate a user with specified username and password. If successful, returns a valid authentication token for the user.
<code>cvauth.get_token</code>	Get a token for a specified user. This method also allows you to exchange a Clearvale network administrator token for a specified user's token.
<code>cvauth.set_cookie</code>	Set a valid session cookie for accessing a Clearvale network.
<code>cvauth.logout</code>	Log out and invalidate authentication tokens for one or all users of the network.
<code>cvauth.is_online</code>	Check if a particular person is logged into Clearvale.
<code>cvauth.renew_token</code>	Renew the expired token for the specified person without needing to ask for the user password or storing it in the application space.

cvauth.authenticate

Authenticate a user with specified username and password. If successful, returns a valid authentication token for the user.

The `auth_token` parameter is not allowed for this method, as this method returns an authentication token.

Access

- This method requires that you use the HTTPS protocol.

Method `POST`

Parameters `username` (required). Username.

`password` (required). Password.

`renewal` (optional). Set this flag to `true` if the application needs to renew the authentication token without requiring to send the user password. This is useful for long running applications and you cannot request the user password. The default value is `false`. An integer that indicates how many days the renewal token is valid. `-1` means to never expire.

`update_token` (optional). Use `true` to get a new token or `false` to use the existing token. The default value is `false`.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.

Returns On success, this method returns the value of the token.

cvauth.get_token

Get a token for a specified user. This method also allows you to exchange a Clearvale network administrator token for a specified user's token.

Access

- This is a privileged method for network admins only. It provides impersonation capability and flexibility for some integration use cases.
- This method does not allow an anonymous caller to call without a token.

Method

GET

Parameters

`user_id` (required). The user ID or login (email address). If set to `-1`, returns a token for a non-logged-in (public) user.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the value of the token.

Example

```
http://network_url/pg/cvapi/rest/xml?method=cvauth.get_token&user_id=1234&auth_token=df202eb08cfaf08f61c6f987b0d68e5e
```

cvauth.set_cookie

Set a valid session cookie for accessing a Clearvale network.

By default, when you set a cookie, the token becomes invalid for increased security. However, there may be cases when you need to use both a token and a cookie. In these cases, use the `from_trusted_agent` parameter to preserve the token.

Method GET

Parameters `from_trusted_agent` (optional). If omitted, the token is invalidated. If set to `true`, preserves the token.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns On success, this method returns `true`.

Examples

```
<iframe  
src="http://network_url/pg/cvapi/rest/xml?method=cvauth.set_cookie&auth  
_token=df202eb08cfaf08f61c6f987b0d68e5e" width="1" height="1"></iframe>
```

```

```

```
<link rel="stylesheet"  
href="http://network_url/pg/cvapi/rest/xml?method=cvauth.set_cookie&aut  
h_token=df202eb08cfaf08f61c6f987b0d68e5e" />
```

The above examples work on all browsers except Internet Explorer 8 or later.

cvauth.logout

Log out and invalidate authentication tokens for one or all users of the network.

Access

- Network members and participants can use this API to log themselves out of the network.
- Network administrators can use this API to log out a particular network user or all users.

Method

GET

Parameters

`user_id` (optional). The user ID or login (email address) of the person to logout. If omitted, and an administrator authentication token is provided, all users of the network are logged out.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.

Returns

On success, this method returns true.

cvauth.is_online

Check if a particular person is logged into Clearvale.

People who sign into your network, but never log out are considered logged out when they have been idle for the amount of time specified for the network session timeout value. The network administrator sets the session timeout value on the **Admin>Security** page. Alternatively, you can set a session timeout value to use when executing this API by using the `timeout` parameter.

Access

- Network members can use this API to check if another member is logged into the network.

Method

GET

Parameters

`user_id` (required). The user ID or login (email address) of the person to check if they are logged in.

`timeout` (optional). The number of minutes that a person can remain idle before they are considered logged out. If you omit this parameter, the session timeout value for the network is used.

Status Code

Code	Description
6	No session timeout value found. To workaround this problem, use the <code>timeout</code> parameter.
0	Success. Any other value indicates that the method failed.
-1	Failed.

Returns

On success, this method returns true.

cvauth.renew_token

Renew the expired token for the specified person without needing to ask for the user password or storing it in the application space.

Access

- This method allows an anonymous caller to call without a token.
- You must be a UNIX administrator to use this method.

Method

GET

Parameters

`user_id` (required). The user ID or login (email address) of the person to renew.

`expired_token` (required). Old token to renew.

`request_tstamp` (required). The current time in milliseconds. This is required to generate the `renewal_code`. In PHP, use the `time()` function to generate this value.

`renewal_code` (required). The renewal code. To get this value, you need to use the following:

```
hash (user_id + request_tstamp, ClientSessionId)
```

where `ClientSessionId` is

```
hash (user_id.token, ClientSharedSecret)
```

Status Code

Code	Description
4	Invalid renewal token.
0	Success. Any other value indicates that the method failed.
-1	Invalid token, including expired token.

Returns

On success, returns the value of the token.

Documents

Get list of documents for a category or community, create documents, delete documents, get document history and so on.

NOTE: “Documents” refer to files in Clearvale; videos, .jpg, .doc, .xls files, and so on can all be uploaded to Clearvale as files.

Method	Description
documents.get_top_folders_in_network	Get list of top folders in current network.
documents.get_top_folders_in_community	Get list of top folders in a community.
documents.get_top_folders	Get list of top folders in current user's personal space.
documents.get_sub_folders	Get list of subfolders.
documents.create_top_folder_in_network	Create a top folder in the current network.
documents.create_top_folder_in_community	Create a top folder in a community.
documents.create_top_folder	Create a top folder in current user's personal space.
documents.create_sub_folder	Create a subfolder.
documents.update_folder	Update either a top-level folder or subfolder.
documents.delete_folder	Delete a folder. Can be either a top-level folder or subfolder.
documents.add_to_folder	Add document to a folder.
documents.remove_from_folder	Remove document from a folder.
documents.get_documents_in_network	Get list of documents in current network.
documents.get_uncategorized_documents_in_network	Get list of uncategorized documents in current network.
documents.get_documents_in_community	Get list of documents in a community.
documents.get_uncategorized_documents_in_community	Get list of uncategorized documents in a community.
documents.get_documents	Get list of documents in current user's personal space. This includes documents owned by the current user and the documents shared (directly) with the current user.
documents.get_uncategorized_documents	Get list of uncategorized documents in current user's personal space.
documents.get_my_documents	Get list of documents owned by current user.
documents.get_shared_documents	Get list of documents shared with current user. This includes documents owned by other users but shared directly with current user or shared indirectly through network membership or community membership.
documents.get_all_documents	Get list of documents accessible to current user. This includes documents owned by current user, documents shared directly with current user, and documents shared indirectly through community membership or network membership.
documents.get_documents_in_folder	Get list of documents in a folder.
documents.get_comments_by_document	Get the comments for a document.
documents.create_document	Create a document.
documents.delete_document	Delete a document.
documents.update_document	Update a document's title and desc attributes.
documents.upload_file	Update document with a new version of file.

Method	Description
documents.download_document	Download a specific version of a document.
documents.download_file	Download the file of a specified version of a document.
documents.get_history	Get document history that consists of a list of events including: version, tstamp, action, user and notes.
documents.share	Share a document with a network, a community, or a member for access.
documents.unshare	Unshare a document with a network, community, or member.
documents.create_external_content	Create new external content. External content is typically a link to an object in an external system.
documents.update_external_content	Update external content.
documents.delete_external_content	Mark external content as deleted.
documents.undelete_external_content	Mark external content as no longer deleted.
documents.set_owner	Change the owner for one or more documents.

documents.get_top_folders_in_network

Get list of top folders in current network.

Access

- This method allows an anonymous caller to call without a token. If you call as an anonymous caller, you can only retrieve folders that are open to the general public.
- Network members can get the top folders in the network.
- If you have a hybrid network, guests must have access to the network space to use this API to get the top folders in the network.

Method

GET

Parameters

None.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the following in an array. Each record includes:

folder_id
folder_name
time_created
time_updated

documents.get_top_folders_in_community

Get list of top folders in a community.

Access

- This method allows an anonymous caller to call without a token. If you call as an anonymous caller, you can only retrieve folders that are open to the general public.
- Network members can get the folders for communities of which they are a member.
- If you have a hybrid network, guests can only use this API for communities of which they are a member.

Method

GET

Parameters

`community_id` (required). Valid community ID specifying a community.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the following in an array. Each record includes:

folder_id
folder_name
time_created
time_updated

documents.get_top_folders

Get list of top folders in current user's personal space.

Access

- Network members can get the top folders in their personal space.
- If you have a hybrid network, guests can use this API to get the top folders in their personal space.

Method

GET

Parameters

None.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the following in an array. Each record includes:

folder_id
folder_name
time_created
time_updated

documents.get_sub_folders

Get list of subfolders.

Access

- This method allows an anonymous caller to call without a token. If you call as an anonymous caller, you can only retrieve folders that are open to the general public.
- Network members can get the folders from the network or communities to which they belong or can view in the network. They can also get subfolders from any user that they can see in the network.
- If you have a hybrid network, guests can use this API to get folders from communities to which they belong. Additionally, if they have access to the network page, they can get subfolders from the network space.

Method

GET

Parameters

`folder_id` (required). Parent folder ID.

`sort_by` (optional). The order in which to sort the returned subfolders. Options are:

- `title`. The folder name. This is the default value.
- `time_created`. The time the folder was created.
- `time_updated`. The time the folder was last updated.

`limit` (optional). Specifies the maximum number of subfolders to return. The default value is 10.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the subfolders of the top folder in an array. Each record includes:

<code>folder_id</code>
<code>folder_name</code>
<code>time_created</code>
<code>time_updated</code>

documents.create_top_folder_in_network

Create a top folder in the current network.

Access

- Only the network administrators can create folders for the network.
- If you have a hybrid network, guests cannot use this API.

Method

POST

Parameters

`title` (required). The folder title.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the ID for the folder.

documents.create_top_folder_in_community

Create a top folder in a community.

Access

- Only community administrators can create folders in a community.
- If you have a hybrid network, guests can use this API to create folders in communities of which they are an administrator.

Method

POST

Parameters

`community_id` (required). Valid community ID specifying a community.

`title` (required). The folder title.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the ID for the folder.

documents.create_top_folder

Create a top folder in current user's personal space.

Access

- Network members can use this API to create a top folder in their personal space.
- If you have a hybrid network, guests can use this API to create a top folder in their personal space.

Method

POST

Parameters

`title` (required). The folder title.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the ID for the folder.

documents.create_sub_folder

Create a subfolder.

Access

- Only network administrators and folder owners can create subfolders.
- If you have a hybrid network, guests can use this API to create subfolders for any folders they own. If the guest is a community administrator, they can create subfolders for the community.

Method

POST

Parameters

`folder_id` (required). The folder ID.

`title` (required). The subfolder title.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the ID for the folder.

documents.update_folder

Update either a top-level folder or subfolder.

Access

- Only network administrators and folder owners can update folders.
- If you have a hybrid network, guests can update any folder that they can edit.

Method

POST

Parameters

`folder_id` (required). The folder ID.

`title` (required). The folder title.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns `1`. If it failed, it returns `0`.

documents.delete_folder

Delete a folder. Can be either a top-level folder or subfolder.

NOTE: If you delete a top-level folder, its subfolders will become top-level folders.

Access

- Only network administrators and folder owners can delete folders.
- If you have a hybrid network, guests can delete any folder that they can edit.

Method

POST

Parameters

`folder_id` (required). The folder ID.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns `1`. If it failed, it returns `0`.

documents.add_to_folder

Add document to a folder.

Access

- Only network administrators and folder owners can add documents to a folder.
- If you have a hybrid network, guests can add documents to any folder that they can edit.

Method

POST

Parameters

`document_id` (required). Document ID.

`folder_id` (required). Folder ID.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns `1`. If it failed, it returns `0`.

documents.remove_from_folder

Remove document from a folder.

Access

- Only network administrators and folder owners can remove documents from a folder.
- If you have a hybrid network, guests can remove any document that they can edit.

Method

POST

Parameters

`document_id` (required). Document ID.

`folder_id` (required). Folder ID.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns `1`. If it failed, it returns `0`.

documents.get_documents_in_network

Get list of documents in current network.

Access

- This method allows an anonymous caller to call without a token. If you call as an anonymous caller, you can only retrieve documents that are visible to the general public.
- Network members can get a list of all documents that they have permission to view.
- If you have a hybrid network, guests can only get a list of documents if they have access to the network space.

Method

GET

Parameters

`limit` (optional). Specifies the maximum number of documents to return. When not specified or set to 0, it returns all.

`offset` (optional). The offset position.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the documents in an array. Each record includes:

<code>id</code>	Document ID
<code>title</code>	Document name
<code>desc</code>	Document description
<code>IsViewOnly</code>	Whether the document can be downloaded or not
<code>time_created</code>	Time document was created
<code>time_updated</code>	Time document was last updated
<code>owner_id</code>	Owner of the document
<code>version</code>	Version of the document
<code>size</code>	Size of the document

documents.get_uncategorized_documents_in_network

Get list of uncategorized documents in current network.

Access

- This method allows an anonymous caller to call without a token. If you call as an anonymous caller, you can only retrieve documents that are visible to the general public.
- Network members can get a list of all documents that they have permission to view.
- If you have a hybrid network, guests can only get documents if they have access to the network space.

Method

GET

Parameters

`limit` (optional). Specifies the maximum number of documents to return. When not specified or set to 0, it returns all.

`offset` (optional). Offset the position.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the documents in an array. Each record includes:

id	Document ID
title	Document name
desc	Document description
IsViewOnly	Whether the document can be downloaded or not
time_created	Time document was created
time_updated	Time document was last updated
owner_id	Owner of the document
version	Version of the document
size	Size of the document

documents.get_documents_in_community

Get list of documents in a community.

Access

- This method allows an anonymous caller to call without a token. If you call as an anonymous caller, you can only retrieve documents that are visible to the general public.
- Network members can get documents from the communities to which they belong.
- If you have a hybrid network, guests can get documents from the communities to which they belong.

Method

GET

Parameters

`community_id` (required). Valid community ID specifying a community.

`limit` (optional). Specifies the maximum number of documents to return. When not specified or set to 0, all returned.

`offset` (optional). The offset position.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the documents in an array. Each record includes:

<code>id</code>	Document ID
<code>title</code>	Document name
<code>desc</code>	Document description
<code>IsViewOnly</code>	Whether the document can be downloaded or not
<code>time_created</code>	Time document was created
<code>time_updated</code>	Time document was last updated
<code>owner_id</code>	Owner of the document
<code>version</code>	Version of the document
<code>size</code>	Size of the document

documents.get_uncategorized_documents_in_community

Get list of uncategorized documents in a community.

Access

- This method allows an anonymous caller to call without a token. If you call as an anonymous caller, you can only retrieve documents that are visible to the general public.
- Network members can get documents from the communities to which they belong.
- If you have a hybrid network, guests can get uncategorized documents from the communities to which they belong.

Method

GET

Parameters

`community_id` (required). Valid community ID specifying a community.

`limit` (optional). Specifies the maximum number of documents to return. When not specified or set to 0, all returned.

`offset` (optional). Offset position.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the documents in an array. Each record includes:

id	Document ID
title	Document name
desc	Document description
IsViewOnly	Whether the document can be downloaded or not
time_created	Time document was created
time_updated	Time document was last updated
owner_id	Owner of the document
version	Version of the document
size	Size of the document

documents.get_documents

Get list of documents in current user's personal space. This includes documents owned by the current user and the documents shared (directly) with the current user.

Access

- Network members can get the documents from their personal space.
- If you have a hybrid network, guests can get documents from their personal space.

Method

GET

Parameters

`limit` (optional). Specifies the maximum number of documents to return. When not specified or set to 0, it returns all.

`offset` (optional). Offset position.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the documents in an array. Each record includes:

<code>id</code>	Document ID
<code>title</code>	Document name
<code>desc</code>	Document description
<code>IsViewOnly</code>	Whether the document can be downloaded or not
<code>time_created</code>	Time document was created
<code>time_updated</code>	Time document was last updated
<code>owner_id</code>	Owner of the document
<code>version</code>	Version of the document
<code>size</code>	Size of the document

documents.get_uncategorized_documents

Get list of uncategorized documents in current user's personal space.

Access

- Network members can get the documents from their personal space.
- If you have a hybrid network, guests can get documents from their personal space.

Method

GET

Parameters

limit (optional). Specifies the maximum number of documents to return. When not specified or set to 0, it returns all.

offset (optional). Offset position.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the documents in an array. Each record includes:

id	Document ID
title	Document name
desc	Document description
IsViewOnly	Whether the document can be downloaded or not
time_created	Time document was created
time_updated	Time document was last updated
owner_id	Owner of the document
version	Version of the document
size	Size of the document

documents.get_my_documents

Get list of documents owned by current user.

Access

- Network members can get the documents that they own.
- If you have a hybrid network, guests can get documents that they own.

Method

GET

Parameters

limit (optional). Specifies the maximum number of documents to return. When not specified or set to 0, it returns all.

offset (optional). Offset position.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the documents in an array. Each record includes:

id	Document ID
title	Document name
desc	Document description
IsViewOnly	Whether the document can be downloaded or not
time_created	Time document was created
time_updated	Time document was last updated
owner_id	Owner of the document
version	Version of the document
size	Size of the document

documents.get_shared_documents

Get list of documents shared with current user. This includes documents owned by other users but shared directly with current user or shared indirectly through network membership or community membership.

Access

- Network members can get the documents shared with them.
- If you have a hybrid network, guests can get documents shared with them.

Method

GET

Parameters

`limit` (optional). Specifies the maximum number of documents to return. When not specified or set to 0, it returns all.

`offset` (optional). Offset position.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the documents in an array. Each record includes:

<code>id</code>	Document ID
<code>title</code>	Document name
<code>desc</code>	Document description
<code>IsViewOnly</code>	Whether the document can be downloaded or not
<code>time_created</code>	Time document was created
<code>time_updated</code>	Time document was last updated
<code>owner_id</code>	Owner of the document
<code>version</code>	Version of the document
<code>size</code>	Size of the document

documents.get_all_documents

Get list of documents accessible to current user. This includes documents owned by current user, documents shared directly with current user, and documents shared indirectly through community membership or network membership.

Access

- Network members can get all documents accessible to them.
- If you have a hybrid network, guests can get all documents accessible to them.

Method

GET

Parameters

`limit` (optional). Specifies the maximum number of documents to return. When not specified or set to 0, it returns all.

`offset` (optional). Offset position.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the documents in an array. Each record includes:

<code>id</code>	Document ID
<code>title</code>	Document name
<code>desc</code>	Document description
<code>IsViewOnly</code>	Whether the document can be downloaded or not
<code>time_created</code>	Time document was created
<code>time_updated</code>	Time document was last updated
<code>owner_id</code>	Owner of the document
<code>version</code>	Version of the document
<code>size</code>	Size of the document

documents.get_documents_in_folder

Get list of documents in a folder.

Access

- This method allows an anonymous caller to call without a token. If you call as an anonymous caller, you can only retrieve documents that are visible to the general public.
- Network members can get documents in folders that they have permission to view.
- If you have a hybrid network, guests can use this API to get documents from folders in communities to which they belong. Additionally, if they have access to the network page, they can get documents from folders in the network space.

Method

GET

Parameters

`folder_id` (required). Folder ID.

`limit` (optional). Specifies the maximum number of documents to return. When not specified or set to 0, it returns all.

`offset` (optional). The offset position.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the documents in an array. Each record includes:

id	Document ID
title	Document name
desc	Document description
IsViewOnly	Whether the document can be downloaded or not
time_created	Time document was created
time_updated	Time document was last updated
owner_id	Owner of the document
version	Version of the document
size	Size of the document

documents.get_comments_by_document

Get the comments for a document.

Access

- This method allows an anonymous caller to call without a token. If you call as an anonymous caller, you can only retrieve comments that are visible to the general public.
- Network members can get comments on documents in folders that they have permission to view.
- If you have a hybrid network, guests can use this API to get comments on documents from folders in communities to which they belong. Additionally, if they have access to the network page, they can get comments on documents from folders in the network space.

Method

GET

Parameters

`document_id` (required). Document ID.

`limit` (optional). Specifies the maximum number of documents to return. When not specified or set to 0, it returns all.

`offset` (optional). The offset from the first match used to go through paginated list of comments when limit parameter limits the number of returned comments. If not specified, 0 is used.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.
7	Invalid or empty document ID.
6	Permission denied.

Returns

On success, this method returns the comments in an array. Each record includes:

id
desc
time_created
time_created_ts
owner

Example

`http://network_url/pg/cvapi/rest/xml?method=blogs.get_comments_by_document&auth_token=7aabb5470e6fadcc610e30c0211d1ccd&document_id=31&limit=10&offset=20`

documents.create_document

Create a document.

A Clearvale document consists of basic attributes and a document file. This method must be called through a multi-part/form data POST request (see RFC1867 for more details on that) that must include the parameters listed below and a file to be uploaded.

Access

- Network members can create a document.
- If you have a hybrid network, guests can use this API to create a document.
- To set a document as preview only (`isViewOnly` set to `1`), your network must have the Document Preview service enabled for it in the Clearvale Management Center.
- Also, you cannot use this API to override network default settings regarding uploading documents and setting a different preference for your document than allowed by the network. That is if the network administrator has configured the network so that all files can be downloaded or previewed, you cannot use the `isViewOnly` set to `1` to override this behavior for a particular file.

Method

POST (multi-part/form data)

Parameters

`title` (required). Title.

`options` (optional). Specify one or more of the following options:

- `desc`. Description.
- `isViewOnly`. Whether the document is downloadable or not. `1` means that the document is view only. `0` means the document can be downloaded.
- `use_html`. One of the following:
 - `yes` indicates that HTML is allowed in the document description. This is the default behavior.
 - `no` indicates that HTML is not allowed in the document description.

Separate the options with the pipe (`|`). For example: `options=desc=annual report|isViewOnly=0`

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the ID for the document.

documents.delete_document

Delete a document.

Access

- Only network administrators and document owners can delete a document.
- If you have a hybrid network, guests can use this API to delete a document.

Method

POST

Parameters

`document_id` (required). Document ID.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns `1`. If it failed, it returns `0`.

documents.update_document

Update a document's `title` and `desc` attributes.

Access

- Only network administrators and document owners can update a document.
- If you have a hybrid network, guests can use this API to update document attributes.

Method

POST

Parameters

`document_id` (required). The document ID.

`title` (required). Title of topic.

`desc` (required). Description of topic.

`options` (optional). Specify the following option:

- `use_html`. One of the following:
 - `yes` indicates that HTML is allowed in the document description. This is the default behavior.
 - `no` indicates that HTML is not allowed in the document description.
- `send_notification`. One of the following:
 - `yes` indicates to allow notifications regarding the updated file.
 - `no` indicates to block notifications regarding the updated file.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns `1`. If it failed, it returns `0`.

documents.upload_file

Update document with a new version of file.

This method must be called through a multi-part/form data POST request (see RFC1867 for more details as well as “[Request Format](#)” on page 14) that must include the parameters listed below and the icon file to be uploaded.

Access

- Network members can upload a document.
- If you have a hybrid network, guests can use this API to upload a document.

Method

POST (multi-part/form data)

Parameters

`document_id` (required). The document ID.

`notes` (required). Notes for current version.

`options` (optional). Specify the following option:

- `send_notification`. One of the following:
 - `yes` indicates to allow notifications regarding the uploaded file.
 - `no` indicates to block notifications regarding the uploaded file.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns `1`. If it failed, it returns `0`.

documents.download_document

Download a specific version of a document.

Access

- This method requires that you use HTTPS.
- This method allows an anonymous caller to call without a token. If you call as an anonymous caller, you can only download documents that are visible to the general public.
- Network members can download any file they have permission to view by providing an authentication token to the list of request parameters.
- If you have a hybrid network, guests can use this API to download documents that they own or that are shared with them.

Method

GET

Parameters

`document_id` (required). Document ID.

`version` (optional). Version number. If not specified, the latest version is returned.

Returns

The HTTP response header returns the following data about the downloaded document:

Content-Type	Document mime type, such as application/msword or application/pdf.
Content-Length	Document size in bytes.
Content-Disposition	Document's file name.
Status	200—Successful. 400—Failure.

On success, the HTTP response body returns the document's contents. On failure, it returns the error.

documents.download_file

Download the file of a specified version of a document.

IMPORTANT: This API is in the process of being retired. Instead, use the API “documents.download_document” on page 110.

Access

- This method allows an anonymous caller to call without a token. If you call as an anonymous caller, you can only download documents that are visible to the general public.
- Network members can download any file they have permission to view.
- If you have a hybrid network, guests can use this API to download documents that they own or that are shared with them.

Method

GET

Parameters

`document_id` (required). Document ID.

`version` (optional). Version number. If not specified, the latest version is returned.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the following:

name	Name of the file.
type	Type of file.
size	Size of file.
data	The file data is encoded within XML, PHP, or JSON.

documents.get_history

Get document history that consists of a list of events including: `version`, `tstamp`, `action`, `user` and `notes`.

Access

- Only network administrators and document owners can get the history for a document.
- If you have a hybrid network, guests can use this API to get the history of documents for which they have permission.

Method

GET

Parameters

`document_id` (required). Document ID.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the following:

<code>version</code>	Version of the file.
<code>tstamp</code>	Timestamp of file.
<code>action</code>	Action to the file.
<code>user</code>	User who took the action.
<code>notes</code>	Any notes specified by the user.

documents.share

Share a document with a network, a community, or a member for access.

Access

- Network members can share a document.
- If you have a hybrid network, guests can use this API to share documents with communities to which they belong or to the network space if they have access.

Method

POST

Parameters

`document_id` (required). Document ID.

`share_id` (required). ID of a network, community, or member with whom to share.

`access` (optional). Either `view` (default) or `edit`.

`options` (optional). Specify the following option:

- `send_notification`. One of the following:
 - `yes` indicates to allow notifications regarding the shared file.
 - `no` indicates to block notifications regarding the shared file.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns `1`. If it failed, it returns `0`.

documents.unshare

Unshare a document with a network, community, or member.

Access

- Network members can unshare a document they have previously shared.
- If you have a hybrid network, guests can use this API to unshare documents they have previously shared.

Method

POST

Parameters

`document_id` (required). Document ID

`share_id` (required). ID of a network, community or member with whom to unshare.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns `1`. If it failed, it returns `0`.

documents.create_external_content

Create new external content. External content is typically a link to an object in an external system.

Access

- Network members can create external content.

Method `POST`

Parameters `title` (required). Document title.

`url` (required). URL of the external object.

`system_type` (required). Must be set to `SHAREPOINT`.

`action_url` (required). External system URL that will perform actions.

`desc` (optional). Optional description of the document.

`options` (optional). Specify the following option:

- `use_html`. One of the following:
 - `yes` indicates that HTML is allowed in the document description. This is the default behavior.
 - `no` indicates that HTML is not allowed in the document description.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns On success, this method returns the ID for the document.

documents.update_external_content

Update external content.

Access • You must be the owner of the external content or have edit permission to edit it.

Method `POST`

Parameters `document_id` (required). The document ID.

`title` (optional). Document title.

`url` (required). URL of the external object.

`system_type` (optional). Must be set to `SHAREPOINT`.

`action_url` (optional). External system URL that will perform actions.

`desc` (optional). Optional description of the document.

`options` (optional). Specify the following option:

- `use_html`. One of the following:
 - `yes` indicates that HTML is allowed in the document description. This is the default behavior.
 - `no` indicates that HTML is not allowed in the document description.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns On success, this method returns `1`. If it failed, it returns `0`.

documents.delete_external_content

Mark external content as deleted.

Access • You must be the owner of the external content or have edit permission to delete it.

Method `POST`

Parameters `document_id` (required). The document ID.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns On success, this method returns `1`. If it failed, it returns `0`.

documents.undelete_external_content

Mark external content as no longer deleted.

Access • You must be the owner of the external content or have permission to undelete it.

Method `POST`

Parameters `document_id` (required). The document ID.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns On success, this method returns `1`. If it failed, it returns `0`.

documents.set_owner

Change the owner for one or more documents.

Access

- Only network administrators can change the owner of a document.

Method

POST

Parameters

`list` (required). The list of document IDs separated by commas.

`owner_id` (required). The new owner's ID or login.

`options` (optional). Specify one or more of the following options:

- `send_notification`. Whether or not to notify the new owner about the document ownership change. Use one of the following values:
 - `yes`: Send email notification to the new document owner. This is the default.
 - `no`: Do not send a notification.
- `cc_folder`. A personal folder of the new owner where they can access the documents. If the folder does not already exist, it is created. You cannot specify a subfolder.

Separate the options with the pipe (`|`). For example:

```
options=send_notification=no|cc_folder=resumes
```

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns a comma-separated list of documents for which the ownership was changed.

Forums

Create forums, delete forums, update forums, delete comments, and so forth. Use these methods for both community and network forums.

NOTE: The `user_id` parameter refers to either a `user_guid` or a username that uniquely identifies a user.

Method	Description
<code>forums.create_topic</code>	Create a forum topic.
<code>forums.delete_topic</code>	Delete a topic.
<code>forums.update_topic</code>	Update a forum topic.
<code>forums.create_comment</code>	Create a comment in a forum topic.
<code>forums.delete_comment</code>	Delete a comment from a topic.
<code>forums.get_comments</code>	Get list of forum topic comments.
<code>forums.get_forums</code>	Get list of forums in a community or in the network.
<code>forums.create_forum</code>	Create a forum in a community or in the network.
<code>forums.delete_forum</code>	Delete a forum.
<code>forums.get_topics</code>	Get list of topics in a forum.

forums.create_topic

Create a forum topic.

Access

- Network members can create forum topics at the network level or within communities to which they belong.
- If you have a hybrid network, guests can use this API to add a forum topic to a forum within guest or external communities to which they belong.

Method

POST

Parameters

`forum_id` (required). Forum ID.

`title` (required). Title of topic.

`desc` (required). Description of the forum topic.

`status` (required). Open or closed.

`options` (optional). Specify the following option:

- `use_html`. One of the following:
 - `yes` indicates that HTML is allowed in the forum description. This is the default behavior.
 - `no` indicates that HTML is not allowed in the forum description.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the forum topic ID.

forums.delete_topic

Delete a topic.

Access

- Network administrators and topic owners can delete a network topic.
- Community administrators and topic owners can delete a community forum topic.
- If you have a hybrid network, guests can use this API to delete a forum topic that they own.

Method

POST

Parameters

`topic_id` (required). Topic ID.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns `1`. If it failed, it returns `0`.

forums.update_topic

Update a forum topic.

Access

- Network administrators and topic owners can update a network topic.
- Community administrators and the topic owner can update a community forum topic.
- If you have a hybrid network, guests can use this API to update a forum topic that they own.

Method

POST

Parameters

`topic_id` (required). Topic ID.

`title` (required). Title of topic.

`desc` (required). Description of topic.

`status` (required). Open or closed.

`options` (optional). Specify the following option:

- `use_html`. One of the following:
 - `yes` indicates that HTML is allowed in the forum topic body. This is the default behavior.
 - `no` indicates that HTML is not allowed in the forum topic body.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the forum topic ID.

forums.create_comment

Create a comment in a forum topic.

Access

- Network members can add comments to forum topics that they can see.
- If you have a hybrid network, guests can use this API to add a comment to a forum topic within guest or external communities to which they belong. Additionally, if they have access to the network page, they can add comments to forum topics in the network space.

Method

POST

Parameters

`topic_id` (required). Topic ID.

`comment` (required). Comment.

`options` (optional). Specify the following option:

- `use_html`. One of the following:
 - `yes` indicates that HTML is allowed in the forum comment. This is the default behavior.
 - `no` indicates that HTML is not allowed in the forum comment.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns `1`. If it failed, it returns `0`.

forums.delete_comment

Delete a comment from a topic.

Access

- Network administrators and comment owners can delete comments.
- Community administrators and comment owners can delete a comments within a community.
- If you have a hybrid network, guests can use this API to delete a forum topic comment that they own.

Method

POST

Parameters

`topic_id` (required). Topic ID.

`comment_id` (required). Comment ID.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns `1`. If it failed, it returns `0`.

forums.get_comments

Get list of forum topic comments.

Access

- This method allows an anonymous caller to call without a token. If you call as an anonymous caller, you can only retrieve public forum comments.
- If you have a hybrid network, guests can use this API to get a comment on a forum topic within guest or external communities to which they belong. Additionally, if they have access to the network page, they can get comments on forum topics in the network space.

Method

GET

Parameters

`topic_id` (required). Topic ID.

`limit` (optional). Specifies the maximum number of comments to return. When not specified or set to 0, it returns all.

`offset` (optional). Offset position.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the following:

<code>id</code>	Comment ID.
<code>message</code>	Comment.
<code>time_created</code>	Time of comment.
<code>owner_id</code>	ID of person who made the comment.
<code>best_answer</code>	Whether or not the particular comment was chosen as the best answer by the forum topic owner. 1 indicates best answer. 0 used for all other comments.

Example

```
http://network_url/pg/cvapi/rest/xml?method=forums.get_comments&auth_token=df202eb08cfaf08f61c6f987b0d68e5e&topic_id=1234
```

forums.get_forums

Get list of forums in a community or in the network.

Access

- This method allows an anonymous caller to call without a token. If you call as an anonymous caller, you can only retrieve public forums.
- If you have a hybrid network, guests can use this API to get forums within guest or external communities to which they belong. Additionally, if they have access to the network page, they can get forums in the network space.

Method

GET

Parameters

`community_id` (required). Community ID specifying a community. 0 indicates network forum topics.

`limit` (optional). Specifies the maximum number of forums to return. When not specified or set to 0, all forums returned.

`offset` (optional). Offset position.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the following:

id
title
desc
time_created
time_updated
forum_order
privacy

forums.create_forum

Create a forum in a community or in the network.

Access

- Only network administrators can create a network forum.
- Network administrators and community administrators can create community forums.
- If you have a hybrid network, guests can use this API to create forums only within guest or external communities of which they are the community administrator.

Method

POST

Parameters

`community_id` (required). Community ID specifying a community. 0 indicates network forum topics.

`title` (required). Title of topic.

`desc` (required). Description of topic.

`privacy` (required). One of the following:

- Network forum setting: `network` or `public`.
- Community forum setting: `network`, `public`, or `community`.
- Internal community forum setting: `community`.
- Guest community forum setting: `community`.
- External community forum setting: `network` or `community`.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the forum ID.

forums.delete_forum

Delete a forum.

Access

- Only network administrators can delete a network forum.
- Network administrators and community administrators can delete community forums.
- If you have a hybrid network, guests can use this API to delete forums that they have created or those that they can edit.

Method

POST

Parameters

`forum_id` (required). Forum ID.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns `1`. If it failed, it returns `0`.

forums.get_topics

Get list of topics in a forum.

Access

- This method allows an anonymous caller to call without a token. If you call as an anonymous caller, you can only retrieve public forum topics.
- If you have a hybrid network, guests can use this API to get forum topics within guest or external communities to which they belong. Additionally, if they have access to the network page, they can get forum topics in the network space.

Method

GET

Parameters

`forum_id` (required). Forum ID.

`limit` (optional). Specifies the maximum number of topics to return. When not specified or set to 0, all topics are returned.

`offset` (optional). Offset position.

`from`. (optional). The beginning date of the date range. Use the format YYYYMMDD. For example, 20160307 is for March 7, 2016.

`to` (optional). The end date of the date range. Use the format YYYYMMDD.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the following:

id
title
desc
status
time_created
owner_id
num_of_comments
last_post_by
last_post_time

Membership

Manage the members, guests, and groups in a network.

Method	Description
membership.get_members_list	Get the list of members.
membership.create_user	Create a new member.
membership.add_member	Add a member to the current network.
membership.update_member	Update a member.
membership.remove_member	Disable a member from the network.
membership.add_member_to_group	Add a person to a Clearvale group.
membership.remove_member_from_group	Remove a member from a Clearvale group.
membership.get_group	Get the details of a Clearvale group, including its members.
membership.create_group	Create a Clearvale group.
membership.delete_group	Delete a Clearvale group.
membership.get_all_groups	Get a list of all Clearvale groups.
membership.get_groups_for_member	Get a list of the Clearvale groups for a particular person.
membership.edit_group	Edit the name of a Clearvale group.
membership.get_following_followers_members_list	Get the list of participants that you are following and the list of participants that are following you.
membership.get_new_members_list	Get the list of new members for a specified time period.

membership.get_members_list

Get the list of members.

Access

- Network members can use this API to get a list of all network members and guests.
- If you have a hybrid network, guests can use this API to get the details of members that are visible to them in the network.

Method

GET

Parameters

`limit` (optional). Specifies maximum number of members to return. When not specified or set to 0, nothing is returned.

`offset` (optional). Default is 0.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the following:

<code>id</code>	User ID.
<code>name</code>	User name.
<code>email</code>	User default email address.
<code>guest_member</code>	1 or 0. 1 means that the user is a guest. 0 means that they are a network member.

Example

```
http://network_url/pg/cvapi/rest/xml?method=membership.get_members_list&auth_token=df202eb08cfaf08f61c6f987b0d68e5e
```

membership.create_user

Create a new member.

After creating a new member, use [membership.add_member](#) to add the member to the current network.

Access

- This method requires that you use the HTTPS protocol.
- This API must be called by a network administrator.

Method

POST

Parameters

`username` (required). The user name (email address) for the new user.

`password` (required). A valid Clearvale password.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
8	Member already exists.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns `1`. If it failed, it returns `0`.

membership.add_member

Add a member to the current network.

The person may already be a member of another Clearvale network or may be new to Clearvale.

Access

- This API must be called by a network administrator or moderator.

Method

POST

Parameters

`email` (required). The email address to use as the login for the new member or the existing member's Clearvale login.

`pwd` (required). The password for the member. (If you set `ignore_password` to 1, set `pwd` to just a placeholder value.)

`ignore_password` (optional). One of the following:

- When passing the parameter `ignore_password = 1`, if the person is new to Clearvale, a temporary password is set for them and they will receive an email to change the password to log in. If the person is already a Clearvale member, they should continue using their own Clearvale password to login to the network.
- 0 indicates to use the password set by the `pwd` parameter.

`firstname` (required). The first name of the member.

`lastname` (required). The last name of the member.

`role` (required). One of the following roles: `administrator`, `moderator`, `writer`, `reader`, or `disabled`. If you have a hybrid network, some additional roles are possible: `external` (for regular guest) or `special_guest`.

`communities` (optional). Community IDs, separated by commas.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns a user ID.

membership.update_member

Update a member.

Access

- This method requires that you use the HTTPS protocol.
- This API must be called by a network administrator or the member being updated.

Method

POST

Parameters

`user_id` (required). The user ID or login (email address).

`options` (optional). Specify one or more of the following options:

- `password`. The new password.
- `firstname`. The new first name.
- `lastname`. The new last name.
- `email`. The new email.
- `role`. The role for the user. Options are:
 - `owner`
 - `administrator`
 - `moderator`
 - `writer`
 - `reader`
 - `external` (use this for regular guests)
 - `special_guest`

Separate the options with the pipe (`|`). For example:

```
options=lastname=Saylor|email=megan.saylor@acme.com
```

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns `1`. If it failed, it returns `0`.

membership.remove_member

Disable a member from the network.

Access

- This API must be called by a network administrator.

Method

GET

Parameters

`user_id` (required). The user ID or login (email address).

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns `1`. If it failed, it returns `0`.

membership.add_member_to_group

Add a person to a Clearvale group.

Access

- For a personal group, this API must be called by the owner of the group.
- For a network group, this API must be called by the network administrator or owner.
- If you have a hybrid network, and a guest creates a personal group, they can only include members and other guests that are visible to them in the network.

Method

POST

Parameters

`user_id` (required). The user ID or login (email address).

`group_id` (required). The personal or network group ID.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
2	Invalid group ID.
3	Invalid user ID.
-20	Failed due to invalid or expired token.

membership.remove_member_from_group

Remove a member from a Clearvale group.

Access

- For a personal group, this API must be called by the owner of the group.
- For a network group, this API must be called by the network administrator or owner.

Method

POST

Parameters

`user_id` (required). The user ID or login (email address).

`group_id` (required). The personal or network group ID.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
2	Invalid group ID.
3	Invalid user ID.
-20	Failed due to invalid or expired token.

membership.get_group

Get the details of a Clearvale group, including its members.

Access

- Network members can use this API to get the details of a personal or network group.
- If you have a hybrid network, guests can use this API to get the details of a personal group.

Method

GET

Parameters

`group_id` (required). The group ID.

`options` (optional). Specify one or more of the following options:

- `sort_by`. The order in which to sort the returned group members. Options are:
 - `name`. Sort by member names.
 - `location`. Sort by member locations.
 - `department`. Sort by member departments.
 - `activitypoints`. Sort by member activity points.

Member name in ascending order is the default. Precede the keyword with `+` to indicate an ascending sort order or `-` to indicate a descending sort order.

- `limit`. The maximum number of group members to return. When not specified, `50` is used.
- `offset`. The offset to use from the first match. If not specified, `0` is used.
- `type`. The type of group to get:
 - `Network`. Get a network group.

- `Personal`. Get a personal group.

If omitted, `personal` is assumed.

Separate the options with the pipe (`|`). For example: `options=sort_by=location|type=network`

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
2	Invalid group ID.
4	Invalid options.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the following:

<code>group_id</code>	Group ID.
<code>group_name</code>	Group name.
<code>members</code>	An array of members.
<code>id</code>	Member ID.
<code>name</code>	Member's display name.
<code>email</code>	Member's default email address (only if called by the network owner or an administrator).
<code>guest_member</code>	Whether the user is a member (0) or guest (1).
<code>location</code>	Member's location.
<code>department</code>	Member's department.
<code>company</code>	Member's company.
<code>activitypoints</code>	Member's activity points.

membership.create_group

Create a Clearvale group.

Access

- For a personal group, this API must be called by a network member.
- For a network group, this API must be called by the network administrator or owner.
- If you have a hybrid network, and a guest creates a personal group, they can only include members and other guests that are visible to them in the network.

Method

POST

Parameters

`groupname` (required). The name of the group.

`type` (required). The type of group, either `personal` or `network`.

`options` (optional). Specify the following option:

- `members`. A comma separated list of user IDs or logins (email addresses) of group members.

For example: `options=members=jsmith@yahoo.com,gturner@gmail.com`

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
4	Invalid options.
5	Unsupported type.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the `group_id` for the new group created.

membership.delete_group

Delete a Clearvale group.

Access

- For personal groups, this API must be called by the group owner.
- For network groups, this API must be called by the network administrator or owner.

Method

POST

Parameters

`group_id` (required). The ID of the group.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
2	Invalid group.
-20	Failed due to invalid or expired token.

membership.get_all_groups

Get a list of all Clearvale groups.

Access

- Network members can access their own personal groups and network groups.
- If you have a hybrid network, guests can get only their personal groups.

Method

GET

Parameters

`options` (optional). Specify one or more of the following options:

- `limit`. The maximum number of groups to return. When not specified, all groups are returned.
- `offset`. The offset to use from the first match. If not specified, 0 is used.
- `type`. The type of groups to get:
 - `Network`. Get network groups.
 - `Personal`. Get personal groups.

If omitted, both network and personal groups are returned.

Separate the options with the pipe (`|`). For example: `options=limit=20|type=personal`

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the following:

network		An array of network groups
	id	Group ID
	name	Group name
personal		An array of personal groups
	id	Group ID
	name	Group name

membership.get_groups_for_member

Get a list of the Clearvale groups for a particular person.

Access

- If you have a hybrid network, guests can only get the personal groups for members and other guests that are visible to them in the network.

Method

GET

Parameters

`user_id` (required). The user ID or login (email address).

`options` (optional). Specify the following option:

- `type`. The type of groups to get:
 - `Network`. Get network groups.
 - `Personal`. Get personal groups.

If omitted, both network and personal groups are returned.

For example: `options=type=personal`

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
3	Invalid user ID.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the following:

network	An array of network groups	
	id	Group ID
	name	Group name
personal	An array of personal groups	
	id	Group ID
	name	Group name

membership.edit_group

Edit the name of a Clearvale group.

Access

- For personal groups, this API must be called by the group owner.
- For network groups, this API must be called by the network administrator or owner.

Method

POST

Parameters

`group_id` (required). The ID of the group.

`options` (optional). Specify the following option:

- `name`. A new name for the group.

For example: `options=name=marketing`

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
2	Invalid group ID.
-20	Failed due to invalid or expired token.

membership.get_following_followers_members_list

Get the list of participants that you are following and the list of participants that are following you.

Alternatively, you can specify another participant's user ID and get the list of participants they are following and a list of participants following them.

Access • Network members and guests can use this API.

Method GET

Parameters `following_limit` (optional). The maximum number of participants you are following to return. If omitted or if 0 is specified, all participants that you are following are returned.

`following_offset` (optional). The offset to use from the first match. If not specified, 0 is used.

`followers_limit` (optional). The maximum number of your followers to return. If omitted or if 0 is specified, all of your followers are returned.

`followers_offset` (optional). The offset to use from the first match. If not specified, 0 is used.

`user_id` (optional). The user ID or login (email address) of a Clearvale participant. Use this option to get the list of participants they are following and a list of participants following them. If `user_id` is not specified, you get the list of participants that you are following and the list of participants that are following you.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.

Returns

On success, this method returns the following:

Following	An array of the people you or the participant specified is following
id	User ID
name	User name
email	Default email address for the user
guest_member	Is the person a network guest? 1 for yes, or 0 for no.
Followers	An array of followers (either yours or the participant specified)
id	User ID
name	User name
email	Default email address for user
guest_member	Is the person a network guest? 1 for yes, or 0 for no

membership.get_new_members_list

Get the list of new members for a specified time period.

Access

- Network members can use this API to get a list of people who joined a network in the time period specified.
- If you have a hybrid network, guests can use this API to get the details of members that are visible to them in the network.

Method

GET

Parameters

from (required). The start date to use. Use either the “y-m-d” or “y/m/d” format.

to (optional). The end date to use. Use either the “y-m-d” or “y/m/d” format. If omitted, the current date is used.

offset (optional). Default is 0.

options (optional). Specify the following option:

- **admin_mode**. Whether to operate in administrative mode or not. Options are **1** for administrative mode and **0** for non-administrative mode. You must be the network owner, supervisor, or moderator to use administrative mode. Note that while you can specify the **admin_mode** option, it does not impact the results returned by the API.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.

Returns

On success, this method returns the following:

id	User ID
name	User name
login_id	Email address for the user
join_date	Date the person joined the network.

Messages

Get information about messages for the current user.

Method	Description
<code>messages.count_unread_messages</code>	Gets the number of unread messages in your Clearvale inbox.
<code>messages.get_message</code>	Get the details of a message.

`messages.count_unread_messages`

Gets the number of unread messages in your Clearvale inbox.

Access

- You are limited to your own messages.

Method `GET`

Parameters None.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns On success, this method returns the number of unread messages.

messages.get_message

Get the details of a message.

Access

- You are limited to your own messages.

Method

GET

Parameters

`message_id` (required). The message ID.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the message details:

<code>id</code>	Message ID.
<code>date</code>	Date when the message was posted.
<code>to</code>	List of recipients.
<code>from</code>	User id/email/name of the poster.
<code>subject</code>	Subject of message.
<code>body</code>	The full message body.

Microblog

Get an existing microblog or create a new one.

Method	Description
<code>microblog.get_microblog_by_microblog_id</code>	Gets a microblog.
<code>microblog.create_microblog</code>	Creates a microblog.

`microblog.get_microblog_by_microblog_id`

Gets a microblog.

Access

- Network members can use this API to get a microblog within the network to which they have access.
- If you have a hybrid network, guests can use this API to get a microblog to which they have access.

Method

GET

Parameters

`microblog_id` (required). The microblog ID.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the microblog.

microblog.create_microblog

Creates a microblog.

Access

- Network members can use this API to create a microblog in any space to which they have access.
- If you have a hybrid network, guests can use this API to create a microblog in any space to which they have access.

Method

POST

Parameters

`desc` (required). The text of the microblog.

`space_id`. The space ID for microblog. Use one of the following options:

- `0` means network space.
- A community ID means a particular community.
- A user ID means a particular member. The default is your user ID.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns `true`.

Org

Create and manage organization structures for your network.

Method	Description
org.import_org	Imports an organization structure from an XML file.
org.add_org	Add an organization node to a system.
org.update_org	Change the name of an organization node.
page.delete_page	Delete an organization node.
org.export_org	Export an organization as an array.

org.import_org

Imports an organization structure from an XML file.

Access

- Network administrators can use this API to import an organization structure.

Method

POST (multi-part/form data)

Parameters

None.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.

Returns

On success, this method returns true to indicate that the organization data was imported.

Example

```
<form action="/pg/cvapi/rest/xml" enctype="multipart/form-data"
method="post">
  <input type="hidden" name="method" value="org.import-org"/>
  file <input type="file" name="org_file" /><br/><br/>
  token <input type="text" name="auth_token" value="" /> <br/><br/>
  <input type="submit" />
</form>
```

The import file should be similar to this, with your own company, division, department, and team names:

```
<?xml version="1.0" encoding="utf-8"?>
<Companies name="acme">
  <Company name="AcmeExports">
    <Division name="Europe">
      <Department name="Marketing">
        <Team name="Social" />
      <Department/>
      <Department name="Sales">
        <Team name="UK" />
      <Department/>
    </Division>
  </Company>
</Companies>
```

org.add_org

Add an organization node to a system.

Access

- Network administrators can use this API to modify the organization structure.

Method

POST

Parameters

`parent_node_path` (required). The tree path of the parent node. Use >>> to separate the parts of the path. The organization structure uses the following hierarchy: companies, divisions, departments, and then teams. For example, if you want to add a team to an existing department, you would enter something like this: `AcmeExports>>>>Europe>>>Marketing`

where:

- `AcmeExports` is a company in your organization
- `Europe` is defined as a division of AcmeExports
- `Marketing` is defined as a department in the Europe division.

`org_level` (required). The level of the organization node where you want to add a new node. Set this to one of the following values:

- `company`
- `division`
- `department`
- `team`.

In the above example of adding a team to the Marketing department, you would specify `team`.

`org_name` (required). The name of the organization node to add.

Concluding the above example, this is where you would specify the team name.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed. Possible reasons for failure are invalid value for <code>parent_node_path</code> or specifying an <code>org_level</code> that doesn't match up with the <code>parent_node_path</code> .

Returns

On success, this method returns true to indicate that the node was added.

org.update_org

Change the name of an organization node.

Access

- Network administrators can use this API to modify the organization structure.

Method

POST

Parameters

`parent_node_path` (required). The tree path of the parent node. Use >>> to separate the parts of the path. The organization structure uses the following hierarchy: companies, divisions, departments, and then teams. For example, if you want to change the name of a time a team, you would enter something like this: `AcmeExports>>>>Europe>>>Marketing`

`org_level` (required). The level of the organization node that you want to change. Set this to one of the following values:

- `company`
- `division`
- `department`
- `team`.

`org_old_name` (required). The name of the organization node to update.

`org_new_name` (required). The new name for the organization node.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed. Possible reasons for failure are invalid value for <code>parent_node_path</code> or specifying an <code>org_level</code> that doesn't match up with the <code>parent_node_path</code> or a wrong <code>old_org_name</code> .

Returns

On success, this method returns true to indicate that the node was updated.

org.delete_org

Delete an organization node.

Access

- Network administrators can use this API to modify the organization structure.

Method

POST

Parameters

`parent_node_path` (required). The tree path of the parent node. Use >>> to separate the parts of the path. The organization structure uses the following hierarchy: companies, divisions, departments, and then teams. For example, if you want to delete a department, you would enter something like this: `AcmeExports>>>>Europe`

`org_level` (required). The level of the organization node that you want to change. Set this to one of the following values:

- `company`
- `division`
- `department`
- `team`.

`org_name` (required). The name of the organization node to delete.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed. Possible reasons for failure are invalid value for <code>parent_node_path</code> or specifying an <code>org_level</code> that doesn't match up with the <code>parent_node_path</code> or a wrong <code>org_name</code> .

Returns

On success, this method returns true to indicate that the node was deleted.

org.export_org

Export an organization as an array.

Access

- Network administrators can use this API to export an organization structure.

Method

GET

Parameters

`parent_node_path` (optional). Either omit this parameter to export the entire organization structure, or enter the `parent_node_path` for the part of the organization to export. Use `>>>` to separate the parts of the path. The organization structure uses the following hierarchy: companies, divisions, departments, and then teams. For example, export the organization structure of a particular division, you would enter something like this: `AcmeExports>>>>Europe`

`is_recursion` (optional). Use this parameter to determine how much of the organization structure to export:

- Use `true` to export recursively and include the entire organization structure, starting at the level indicated by the `parent_node_path`.
- Use `false` to export just the immediate children of the `parent_node_path`. The children of the children will not be exported. This is the default value.

To export just the company-level organization nodes and not any of the children, omit both the `parent_node_path` and `is_recursion` parameters.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed. Possible reason for failure is an invalid value for <code>parent_node_path</code> .

Returns

On success, this method returns an array that looks as follows:

```
$result = array(
    'id' => 1,
    'name' => 'Broadvision',
    'parent_id' => -1,
    'children' => array(
        [0] => array(
            'id' => 11,
            'name' => 'GC',
            'parent_id' => 1,
            'children' => array(
                [0] => array(
                    'id' => 111,
                    'name' => 'Dev1',
                    'parent_id' => 11,
                    'children' => array()
                ),
            )
        ),
        [1] => array(
            'id' => 12,
            'name' => 'US',
            'parent_id' => 1,
            'children' => array()
        ),
        [2] => array(
            'id' => 13,
            'name' => 'Japan',
            'parent_id' => 1,
            'children' => array()
        ),
    )
);
```

Page

Create, update, delete, and get wiki pages.

Method	Description
<code>page.create_page</code>	Creates a wiki page.
<code>page.update_page</code>	Updates a wiki page.
<code>page.delete_page</code>	Deletes a wiki page.
<code>page.get_sub_pages</code>	Gets the wiki sub-pages for a specific wiki page.
<code>page.get_root_page</code>	Gets the root wiki page for a network, community, or member.
<code>page.get_pages</code>	Gets the wiki pages for a network, community, or member.

page.create_page

Creates a wiki page.

Access

- Network members can use this API to create wiki pages within the network.
- If you have a hybrid network, guests can use this API to create wiki pages on their personal page or within guest or external communities to which they belong. Additionally, if they have access to the network page, they can create wiki pages in the network space.

Method

POST

Parameters

`parent_page_id` (required). The parent wiki page ID.

`title` (required). Wiki page title.

`options` (optional). Specify one or more of the following options:

- `body`. The wiki page content.
- `tags`. A keyword for this wiki page.
- `read_access`. Determines who can see the wiki page. Options are:
 - 0 means creator only.
 - 1 means network members. This is the default value.
 - 2 means general public.
 - 3 means community members.
- `write_access`. Determines who can edit the wiki page. Options are:
 - 0 means creator only. This is the default value.
 - 1 means network members.
 - 3 means community members.
- `use_html`. One of the following:
 - `yes` indicates that HTML is allowed in the wiki page. This is the default behavior.
 - `no` indicates that HTML is not allowed in the wiki page.

Separate the options with the pipe (`|`). For example: `options=body=This is sample wiki text|read_access=2|write_access=0`

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns a page ID.

page.update_page

Updates a wiki page.

Access

- Network members can update their wiki pages.
- If you have a hybrid network, guests can use this API to update wiki pages on their personal page or within guest or external communities to which they belong. Additionally, if they have access to the network page, they can create wiki pages in the network space. In the case of editing wiki pages within a community or in the network space, they must have permission to edit the wiki page.

Method

POST

Parameters

`page_id` (required). The wiki page ID.

`options` (optional). Specify one or more of the following options:

- `title`. The wiki page title.
- `body`. The wiki page content.
- `tags`. A keyword for this wiki page.
- `read_access`. Determines who can see the wiki page. Options are:
 - 0 means creator only.
 - 1 means network members. This is the default value.
 - 2 means general public.
 - 3 means community members.
- `write_access`. Determines who can edit the wiki page. Options are:
 - 0 means creator only. This is the default value.
 - 1 means network members.
 - 3 means community members.
- `use_html`. One of the following:
 - `yes` indicates that HTML is allowed in the wiki page. This is the default behavior.
 - `no` indicates that HTML is not allowed in the wiki page.

Separate the options with the pipe (`|`). For example: `options=body=This is sample wiki text|read_access=2|write_access=0`

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns `1`.

page.delete_page

Deletes a wiki page.

Access

- Network members can delete their wiki pages.
- If you have a hybrid network, guests can use this API to delete wiki pages on their personal page or within guest or external communities to which they belong. Additionally, if they have access to the network page, they can delete wiki pages in the network space. In the case of deleting wiki pages within a community or in the network space, they must have permission to edit the wiki page.

Method

POST

Parameters

`page_id` (required). The wiki page ID.

`recursive`. Whether to delete all sub-level wiki pages. `1` means to delete all sub-levels of pages. `0` moves up the sub-pages. Default is `0`.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns `1`.

page.get_sub_pages

Gets the wiki sub-pages for a specific wiki page.

Access

- This method allows an anonymous caller to call without a token. If you call as an anonymous caller, you can only retrieve public pages. Network members can get any pages that they can see within the network.
- If you have a hybrid network, guests can use this API to get pages that they can see, including on their personal page or within guest or external communities to which they belong. Additionally, if they have access to the network page, they can get wiki pages in the network space.

Method

GET

Parameters

`page_id` (required). The wiki page ID.

`recursive`. Whether to return all levels of wiki pages or just the first level. `1` means to return all levels of pages. `0` returns just the first level page. Default is `0`.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the following:

guid
title
description
container_guid
parent_guid
type
subtype
time_created
time_updated

page.get_root_page

Gets the root wiki page for a network, community, or member.

Access

- This method allows an anonymous caller to call without a token. If you call as an anonymous caller, you can only retrieve public pages.
- Network members can get any pages that they can see within the network.
- If you have a hybrid network, guests can use this API to get pages that they can see, including on their personal page or within guest or external communities to which they belong. Additionally, if they have access to the network page, they can get wiki pages in the network space.

Method

GET

Parameters

`space_id` (required). The space ID for the root page to get. Use one of the following options:

- 0 means network space.
- Community ID means a particular community.
- User ID means a particular member.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the following:

guid
title
description
container_guid
parent_guid
type
subtype
time_created
time_updated

page.get_pages

Gets the wiki pages for a network, community, or member.

Access

- This method allows an anonymous caller to call without a token. If you call as an anonymous caller, you can only retrieve public pages.
- Network members can get any pages that they can see within the network.
- If you have a hybrid network, guests can use this API to get pages that they can see, including on their personal page or within guest or external communities to which they belong. Additionally, if they have access to the network page, they can get wiki pages in the network space.

Method

GET

Parameters

`space_id` (required). The space ID for the wiki pages to get. Use one of the following options:

- `0` means network space.
- Community ID means a particular community.
- User ID means a particular member.
- `-1` means no restriction on spaces.

`options` (optional). Specify one or more of the following options:

- `from`. The earliest date for the pages to get. Use the format YYYYMMDD. For example, 20110407 is for April 7, 2011. If not specified, all pages are returned up to the current date (or the date specified for `to`).
- `to`. The latest date for the pages to get. Use the format YYYYMMDD. If not specified, all pages are returned (or all pages created after the `from` date if that is specified).
- `limit`. The maximum number of pages to return. When not specified, 50 is used.
- `offset`. The offset to use from the first match. If not specified, 0 is used.
- `sort`. Keywords that indicate sort order of the search results. Sort options are: `time_updated`, `time_created`, `name`, `creator_name`, `owner_name`, or `status`. Precede the keyword with `+` to indicate an ascending sort order or `-` to indicate a descending sort order.
- `date_flag`. Whether to get pages based on the creation date or last modified date. Use `0` to indicate creation time and `1` to indicate last modified time. 0 is the default.

Separate the options with the pipe (`|`). For example:

```
options=from=20110701|to=20110731|sort=status -time_updated
```

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the following:

guid
subtype
owner_guid
container_guid
site_guid
access_id
time_created
time_updated
title
description
write_access_id
parent_guid

Poll

Create polls, delete polls, update polls, vote in polls, and so forth. Use these methods for community, network, or personal polls.

Method	Description
<code>poll.get_polls</code>	Gets the polls for the specified date range.
<code>poll.create_poll</code>	Creates a new poll.
<code>poll.update_poll</code>	Updates a poll.
<code>poll.delete_poll</code>	Deletes a poll.
<code>poll.get_poll_results</code>	Gets the results of a poll.
<code>poll.vote</code>	Lets you vote in a poll.
<code>poll.is_voted_by_user</code>	Determines if a particular member voted in a poll.

`poll.get_polls`

Gets the polls for the specified date range.

Access

- Network members can get polls within the network that they have permission to view.
- If you have a hybrid network, guests can use this API to get polls on their personal page or within guest or external communities to which they belong. Additionally, if they have access to the network page, they can get polls in the network space.

Method

GET

Parameters

`space_id` (required). The space within which to get polls. Options are:

- `0` for the network space.
- A community ID for the community to use.
- A user ID for the member to use.

`options` (optional). Specify one or more of the following options:

- `status`. Status of the poll. `0` means closed. `1` means open (default).
- `limit`. The maximum number of polls to return. If not specified, `0` is used, and that returns 50 polls.
- `offset`. The offset from the first match. If not specified, `0` is used.
- `sort`. Keywords that indicate the sort order of the search results. Sort options are: `time_updated`, `time_created`, `name`, `creator_name`, `owner_name`, or `status`. Precede the keyword with `+` to indicate an ascending sort order or `-` to indicate a descending sort order.
- `from`. The beginning date of the date range. Use the format YYYYMMDD. For example, 20110407 is for April 7, 2011. If not specified, all polls are returned up to the current date (or the date specified for `to`).
- `to`. The end date of the date range. Use the format YYYYMMDD. To get all polls, omit the `from` and `to` options. If not specified, all polls are returned (or all polls created after the `from` date if that is specified).

- `date_flag`. Whether to use the poll creation date or the last time the poll was modified to determine the date range. `0` means to use poll creation date and `1` means use last date that the poll was modified. If not specified, `0` is used.

Separate the options with the pipe (`|`). For example: `options=status=0|limit=10`

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the following for each poll:

<code>guid</code>	Poll ID.
<code>subtype</code>	The value 12, which indicates that this is a poll.
<code>owner_guid</code>	ID of person who created poll.
<code>container_guid</code>	Where the poll appears. Can be one of the following: <ul style="list-style-type: none"> • For polls in the network space, returns the value 0. • For polls in a community, returns the community ID. • For polls on a personal page, returns the user ID of the poll owner.
<code>site_guid</code>	Site ID.
<code>access_id</code>	Who can access the poll: <ul style="list-style-type: none"> • 0 indicates a private poll and only you can see it. • 1 indicates a network-level poll. • 2 indicates a public poll.
<code>time_created</code>	Time the poll was created.
<code>time_updated</code>	Time the poll was updated.
<code>title</code>	Title of the poll.
<code>description</code>	Description of the poll.
<code>question</code>	Subject of this poll.
<code>site_selected</code>	site the poll selected, 1 means site selected, 0 means not.
<code>single_selection</code>	<code>yes</code> means single selection, <code>no</code> means more than one selection possible.
<code>vote_once_only</code>	<code>yes</code> means people can vote only once in the poll, <code>no</code> means they can vote multiple times.
<code>status</code>	<code>open</code> means the poll is still open, and <code>closed</code> means it is closed and participants can no longer vote in the poll.
<code>response</code>	An array of the poll options (the possible values that you can choose for the poll).

poll.create_poll

Creates a new poll.

Access

- Network members can create polls within the network.
- If you have a hybrid network, guests can use this API to create polls on their personal page or within guest or external communities to which they belong. Additionally, if they have access to the network page, they can create polls in the network space.

Method

POST

Parameters

`question` (required). The poll subject.

`answer` (required). The poll options. Use commas to separate the values. For example: "yes,no"

`single_selection` (required). Whether or not you can select more than one answer in the poll. `1` means single selection and `0` means multiple selections.

`vote_once_only` (required). Whether or not you can vote multiple times in the poll. `1` means you can vote only one time. `0` means that you can vote multiple times.

`options` (optional). Specify one or more of the following options:

- `status`. Status of the poll. `0` means closed. `1` means open (default).
- `tags`. Tags for search.
- `space_id`. The space in which to create the poll. Options are:
 - A community ID to place the poll in a community.
 - `0` to place the poll at the network level.
 - Omit this parameter to place the poll in the current user's space.
- `privacy`. The privacy for the poll. Use `0` for Only Me, `1` for network members, `2` for General Public, and `3` for community members.

Separate the options with the pipe (`|`). For example: `options=status=1|space_id=0`

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the new poll ID.

poll.update_poll

Updates a poll.

Access

- Network members can update polls that they created within the network.
- If you have a hybrid network, guests can use this API to update any polls that they created.

Method

POST

Parameters

`poll_id` (required). The poll ID.

`options` (optional). Specify one or more of the following options:

- `question`. The poll subject.
- `answer`. The poll options. Use commas to separate the values. For example: "yes,no"
- `single_selection`. Whether or not you can select more than one answer in the poll. `1` means single selection and `0` means multiple selections.
- `vote_only_once`. Whether or not you can vote multiple times in the poll. `1` means you can vote only one time. `0` means that you can vote multiple times.
- `status`. Status of the poll. `0` means closed. `1` means open (default).
- `tags`. Tags for search.
- `privacy`. The privacy for the poll. Use `0` for Only Me, `1` for network members, `2` for General Public, and `3` for community members.

Separate the options with the pipe (`|`). For example: `options=question=Are you using the Clearvale iPhone app?|answer=yes,no|single_selection=1|vote_only_once=1`

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the string "success."

poll.delete_poll

Deletes a poll.

Access

- Network members can delete polls that they created within the network.
- If you have a hybrid network, guests can use this API to delete any polls that they created.

Method

POST

Parameters

`poll_id` (required). The poll ID.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Return

On success, this method returns the string “success.”

poll.get_poll_results

Gets the results of a poll.

Access

- This method allows an anonymous caller to call without a token. If you call as an anonymous caller, you can only retrieve public polls.
- Network members can retrieve results for any poll they have permission to view.
- If you have a hybrid network, guests can use this API to get poll results on their personal page or within guest or external communities to which they belong. Additionally, if they have access to the network page, they can get poll results in the network space.

Method

GET

Parameters

`poll_id` (required). The poll ID.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns an array with the poll results.

poll.vote

Lets you vote in a poll.

Access

- Network members can vote in any poll that they have permission to view.
- If you have a hybrid network, guests can use this API to vote in polls on their personal page or within guest or external communities to which they belong. Additionally, if they have access to the network page, they can vote in polls in the network space.

Method

POST

Parameters

`poll_id` (required). The poll ID.

`values` (required). The poll answer. If the poll allows multiple selections, use commas to separate the answers. For example: "`monday,wednesday,friday`"

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the string "success."

poll.is_voted_by_user

Determines if a particular member voted in a poll.

Access

- This method allows an anonymous caller to call without a token. If you call as an anonymous caller, you can only retrieve public poll information.
- Network members can use this method on polls that they created.
- If you have a hybrid network, guests can use this API on any polls that they created.

Method

GET

Parameters

`poll_id` (required). The poll ID.

`user_id` (required). The user ID.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns `true` if the member voted in the poll or `false` if they did not vote.

Profile

Update profile attributes, get user profile, and so on.

NOTE: The `user_id` parameter refers to either a `user_guid` or a username that uniquely identifies a user.

Method	Description
<code>profile.get_profile_for_user</code>	Gets the profile for specified user.
<code>profile.update_user_profile</code>	Updates an attribute of the specified user profile.
<code>profile.update_user_name</code>	Updates the name of a user.
<code>profile.get_recognitions</code>	Get recognition information for participants of the network.
<code>profile.upload_profile_icon</code>	Upload a new profile picture.

`profile.get_profile_for_user`

Gets the profile for specified user.

Access

- Network members can use this API to get information on other network participants.
- If you have a hybrid network, guests can use this API to get the profile for any network participants that they can see in the network.

Method

GET

Parameters

`user_id` (required). The user ID or login (email address).

`options` (optional). Specify the following option:

- `admin_mode`. Whether to operate in administrative mode or not. Options are `1` for administrative mode and `0` for non-administrative mode. You must be the network owner, supervisor, or moderator to use administrative mode. Use administrative mode to retrieve the user's role.

For example: `options=admin_mode=1`

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the following for the user:

id	User ID.
name	Name of member. If the user has a display name specified, that is returned. If not, the user's name is used.
firstname	The user's first name. If a display name appears for <code>name</code> , <code>firstname</code> can be null.
lastname	The user's last name. If a display name appears for <code>name</code> , <code>lastname</code> can be null.
email	Default email address.
language	Default language.
code	User's code.
banned	Has the person been disabled within in Clearvale? (yes or no).
last_login	Date of last login.
previous_last_login	Date of previous last login.
last_action	Date of last action in the network.
username	Login email.
icon	URL for profile picture.
time_created	Time when the user was created within Clearvale.
time_updated	Time when the user was last updated within Clearvale.
role	The user's role. Only administrators can see the role for a user. For all others, null is returned.
validated	Whether the user has been validated.
validated_method	How the person validated their network membership. Possible values are <code>email</code>
i18n_patterns	
has_icon	Profile image is set.
mobile_saved_files_partial_tree	
mobile_my_folder_full_tree	
mobile_folder_name_dictionary	
ecosystem_logged_in	
admin_role_col	
briefdescription	Title.
companyname	Company name.
department	Department.
contactemail	Email address.
mobile_email	Mobile email address.
phone	Phone number.
mobile	Mobile phone number.
location	Location.
country	Country.

interests	Interests.
skills	“Specialized in’ field in user profile.
website	Personal website.
skypeid	Skype ID.
msnid	Microsoft ID
yahoid	Yahoo ID.
aolimid	AOL ID.
description	“Current Projects” field in user profile.
admin	
admin_created	
icontime	Time profile image was set.
x1	
x2	
y1	
y2	
guest_member	Is the person a network guest? 1 for yes, and 0 for no.

Example

`http://network_url/pg/cvapi/rest/xml?method=profile.get_profile_for_user&auth_token=df202eb08cfaf08f61c6f987b0d68e5e&user_id=jane.lewis@broadvision.com`

profile.update_user_profile

Updates an attribute of the specified user profile.

Access

- Network members can only update their own profile. Network administrators can pass the `user_id` of another participant.
- If you have a hybrid network, guests can use this API to update their own profile.

Method

POST

Parameters

`user_id` (required). The user ID or login (email address).

`name` (required). Profile attribute name to update. The possible attributes to update are:

Attribute	Corresponding Profile Field	Notes
briefdescription	Title	Use this attribute for job title.
companyname	Company	
division	Division	
department	Dept.	
team	Team	
contactemail	Email	
mobileemail	Mobile Email	
phone	Phone	
mobile	Mobile	
location	Location	
country	Country	
interests	Interests	
skills	Specialized in	
website	My Website	
skypeid	Skype ID	
msnid	MSN	
yahoid	Yahoo	
aolimid	AOL IM	
description	Current Projects	

If your network has custom profile fields specified, you can update them as well.

`value` (required). New value for profile attribute.

`privacy` (required). Who can see the profile attribute. Options are: Use `0` for Only Me, `1` for network members, and `2` for General Public. If network administrators are setting profile attributes for another person, they cannot set `privacy` to `0`.

`force` (optional). Set to `yes` to indicate to use any values set for profile attributes, including blank values to force parameter values to be blank. The `force` parameter is optional.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns On success, this method returns a success message.

profile.update_user_name

Updates the name of a user.

Access

- Network members can only update their own name. Network administrators can pass the `user_id` of another participant.
- If you have a hybrid network, guests can use this API to update their own name.

Method POST

Parameters `user_id` (required). The user ID or login (email address).

`firstname` (required). The new first name for the user.

`lastname` (required). The new last name for the user.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns On success, this method returns `1`.

profile.get_recognitions

Get recognition information for participants of the network.

Access

- Network members can use this API to get information on other network participants.
- If you have a hybrid network, guests can use this API to get recognitions for any network participants that they can see in the network.

Method

GET

Parameters

`options` (optional). Specify one or more of the following options:

- `limit`. The maximum number of recognitions to return. When not specified, `20` is used.
- `offset`. The offset to use from the first match. If not specified, `0` is used.
- `user_id`. The user ID or login (email address). If omitted, recognition for the current user is returned.
- `from`. The earliest date for the recognitions to get. Use the format `YYYYMMDD`. For example, `20110407` is for April 7, 2011. If not specified, all recognitions are returned up to the current date (or the date specified for `to`).
- `to`. The latest date for the recognitions to get. Use the format `YYYYMMDD`. If not specified, all recognitions are returned (or all recognitions created after the `from` date if that is specified).

Separate the options with the pipe (`|`). For example:

```
options=user_id=36|from=20120701|to=20130731
```

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
1	Invalid date format or a from date earlier than the to date.

Returns

On success, this method returns the following in an array. Each record includes:

user_id	Receiver of recognition
sender_id	Sender of recognition
timestamp	When the recognition was sent
message	Message associated with the recognition icon
gift	One of the following: <ul style="list-style-type: none">• highfive• smile• star• thumbsup
privacy	One of the following values: <ul style="list-style-type: none">• public. People see the recognition icon and message• private. People can see the recognition icon but not the message

profile.upload_profile_icon

Upload a new profile picture.

This method must be called through a multi-part/form data POST request (see RFC1867 for more details) that must include the parameter listed below and the profile picture file to be uploaded.

Access

- Network members can use this API to upload a new profile picture for themselves.
- Network administrators can use this API to upload a picture for other network participants.
- If you have a hybrid network, guests can use this API to upload a profile picture.

Method

POST (multi-part/form data)

Parameters

`user_id` (required). The user ID or login (email address).

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to an invalid or expired token.

Returns

On success, this method returns `1`. It returns `0` if it failed.

Search

Search a network.

Method	Description
search.search	Searches a network and returns matching content.

search.search

Searches a network and returns matching content.

Access

- This method allows an anonymous caller to call without a token. If you call as an anonymous caller, you can only retrieve public information.
- Network members can use this method to find content visible to them within the network.
- For a hybrid network, can guests search for any content that is visible to them within the network.

Method

GET

Parameters

`search_key` (required). The search keyword.

`options` (optional). Specify one or more of the following options:

- `type`. One of the following values:
 - `member`. Return matching users.
 - `community`. Return matching communities.
 - `blog`. Search just blogs.
 - `bookmark`. Search just bookmarks.
 - `embedded`. Search just embedded files.
 - `file`. Search just files.
 - `event`. Search just events.
 - `topic`. Search just forum topics.
 - `poll`. Search just polls.
 - `wiki`. Search just wiki pages.
 - `all`. Search all content. This is the default value.
- `filetype`. This parameter is used for subtypes of `file` or `embedded` only. Possible values are:
 - `document`. Search all document files.
 - `picture`. Search all image files.
 - `video`. Search all video files.
 - `audio`. Search all audio files.
 - `general`. Search all other types of files.
 - `all`. Search all files. This is the default value.

- `community_id`. The string identifying the communities to search. Use commas to separate multiple community identifiers. For example: `23,24,25`. To search all communities, use the keyword `all`.
- `sort`. A string that indicates sort order of the search results:
 - `+time_updated`. Sort in ascending order based on when the item was last updated.
 - `-time_updated`. Sort in descending order based on when the item was last updated.
 - `+time_created`. Sort in ascending order based on when the item was created.
 - `-time_created`. Sort in descending order based on when the item was created.
 - `+name`. Sort in ascending order based on when the name of the item.
 - `-name`. Sort in descending order based on when the name of the item.
- `limit`. The maximum number of matches to return. If not specified, `50` is used.
- `offset`. The offset from the first match. If not specified, `0` is used.

Separate the options with the pipe (`|`). For example:

```
options=type=file|filetype=picture|sort=+time_created
```

Status

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
2	No results found.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the following:

<code>id</code>	The ID for the object that is returned.
<code>name</code>	The name of the object that is returned.
<code>type</code>	The type of the returned object.
<code>url</code>	The URL for the returned object.
<code>last_updated_time</code>	The last time the returned object was updated.

Example

To search your network for a blog with the word `safety`:

```
http://network_url/pg/cvapi/rest/xml?method=search.search&search_key=safety&options=type=blog
```

To search a particular community for a Word file that contains the word `safety`:

```
http://network_url/pg/cvapi/rest/xml?method=search.search&search_key=safety&options=type=file|filetype=document|community_id=23
```

Statistics

Get statistics pertaining to your Clearvale network.

Method	Description
<code>statistics.get_network_statistics</code>	Get the network statistics that appear on the Admin>Statistics page.
<code>statistics.get_member_reports</code>	Get the member reports for a particular user.

statistics.get_network_statistics

Get the network statistics that appear on the **Admin>Statistics** page.

Access

- Only network administrators can get network statistics.

Method

GET

Parameters

`from`. (optional). The start date for the statistics to get. Use the format YYYYMMDD. For example, 20110407 is for April 7, 2011. If not specified, all statistics are returned up to the current date (or the date specified for `to`).

`to` (optional). The end date for the statistics to get. Use the format YYYYMMDD. If not specified, all statistics are returned (or all statistics created after the `from` date if that is specified).

Status Code

2

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.
1	Invalid date format, including the from date might be earlier than the to date.
2	User is not an administrator.

Returns

On success, this method returns the following in an array. Each record includes:

<code>total_members</code>	Total number of network members. If you have a hybrid network, this does not include any guests.
<code>total_guests</code>	Total number of people with the guest role (not including special guests)
<code>total_special_guests</code>	Total number of people with the special guest role.
<code>total_participants</code>	Total number of network members and all guests.
<code>total_users</code>	Total number of network members, all guests, and any disabled members.

networks	Total number of networks (may be more than 1 if the network is part of an ecosystem).
communities	Total number of communities.
internal_communities	If you have a hybrid network, number of internal communities.
guest_communities	If you have a hybrid network, number of guest communities.
external_communities	If you have a hybrid network, number of external communities.
role	Number of member roles in the network.
file	Number of files in the network.
dvcats	Number of folders in the network.
widget	Number of widgets used in the network.
pages_welcome	Number of root wiki pages in the network.
page_top	Number of top-folder wiki pages in the network.
page	Number of wiki pages in the network.
external_content	Number of external links in the network.
bfolder	Number of blog folders in the network.
event_calendar	Number of events defined in the network.
poll	Number of polls in the network.
gfolder	Number of community folders in the network.
task	Number of tasks in the network.
ufolder	Number of groups in the network, including personal and network-level.
blog	Number of blog posts in the network.
messages	Number of messages sent in the network.
bookmarks	Number of bookmarks set in the network.
forumtopic	Number of forum topics in the network.
forum	Number of forums in the network.
task_notification	Number of task notifications in the network.
microblog	Number of status updates specified in the network.

statistics.get_member_reports

Get the member reports for a particular user.

Access

- Network members can see the activity reports for other members.

Method

GET

Parameters

`user_id`. (required). ID of the user.

`options` (optional). Specify a string that contains one of the following:

- `from`: The earliest date for the report to get. Use the format YYYYMM (201304 for April, 2015). If not specified, the current week is considered.
- `to`: The latest date for the report to get. Use the format YYYYMM. If not specified, all reports are returned up to the current date. If the `from` parameter is specified, the `to` parameter must be equal or greater.
- `period`: One of the following: `weekly`, `monthly`, or `quarterly`. Default is `weekly`. If the difference between `to` and `from` is greater than a month, the default will be `monthly`. If the difference is greater than a quarter, the default will be `quarterly`.

Separate the options with the pipe (`|`). For example:

```
options=from=201301|to=201306|period=monthly
```

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
1	Invalid date format or <code>from</code> date earlier than <code>to</code> date.

Returns

On success, this method returns the following in an array. Each record includes:

<code>from</code>	Beginning of the period.
<code>to</code>	End of period.
<code>activity_points</code>	Total activity points.
<code>popularity_points</code>	Total popularity points.
<code>connectivity_points</code>	Total connectivity points.

Task

Create tasks, delete tasks, update tasks, and so forth.

Method	Description
<code>task.create_task</code>	Creates a task.
<code>task.create_task_with_options</code>	Creates a task.
<code>task.update_task</code>	Updates a task.
<code>task.delete_task</code>	Deletes a task.
<code>task.get_assigned_tasks</code>	Gets the tasks assigned to yourself.
<code>task.get_created_tasks</code>	Gets the tasks created by a specific member.
<code>task.get_comments_of_task</code>	Gets the comments for a specific task.
<code>task.add_comment_to_task</code>	Adds a comment to a task.
<code>task.delete_comment_from_task</code>	Deletes a comment from a task.
<code>task.add_step</code>	Adds a step to a task.
<code>task.update_step</code>	Updates a step within a task.
<code>task.delete_step</code>	Deletes a step within a task.
<code>task.reorder_steps</code>	Changes the order of the steps that belong to a task.
<code>task.complete_task_assigned_to_me</code>	Completes the task or task step assigned to you.
<code>task.accept_task_assigned_to_me</code>	Accepts the task or task step assigned to you.
<code>task.decline_task_assigned_to_me</code>	Declines the task or task step assigned to you.
<code>task.acknowledge_task_notification</code>	Accept acknowledgement that a task has been completed.
<code>task.get_step_assignees</code>	Get the assignees for a task step.
<code>task.get_task_by_id</code>	Get task by ID.
<code>task.add_attachments</code>	Add attachments to a task.
<code>task.remove_attachments</code>	Remove attachments from a task.
<code>task.add_assignees</code>	Add assignees to a task.
<code>task.remove_assignees</code>	Remove assignees from a task.

task.create_task

Creates a task.

Access

- Network members can use this API to create a task.
- If you have a hybrid network, guests can use this API to create a task for anyone in the network that they can see and that is named to them.

Method

POST

Parameters

`task_name` (required). The name of the task.

`assigned_members` (required). One or more user IDs. Separate multiple user IDs with commas (,).

`options` (optional). Specify one or more of the following options:

- `due_date`. The date the task is due. Use the format YYYYMMDD. For example, 20110407 is for April 7, 2011.
- `description`. A description for the task.
- `attachments`. One or more attachment IDs. Separate multiple IDs with commas (,).
- `action_type`. The type of task. Options are:
 - `none`. No type specified. This is the default value.
 - `approval`. An approval task. The task assignee will need to approve this task. These tasks have slightly different states than other tasks (approved instead of completed).
 - `consultation`. Use this for tasks where you are requesting consultation help.
 - `directive`. Use this for tasks where you direct the assignee to do something.
 - `invitation`. Use this for tasks where you invite the assignee to an event.
 - `review`. Use this for tasks where you ask assignee to review something.
- `flag`. The importance of the task. Options are:
 - `important`. The task importance flag turns orange when this value is set.
 - `normal`. Default value.
- `execute_mode`. Whether or not all assigned members need to complete the task or just one member. Options are `all` or `only_one`.
- `use_html`. One of the following:
 - `yes` indicates that HTML is allowed in the task description. This is the default behavior.
 - `no` indicates that HTML is not allowed in the task description.

Separate the options with the pipe (|). For example:

```
options=due_date=20110407|action_type=consultation|execute_mode=only_one
```

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the ID of the new task.

task.create_task_with_options

Creates a task.

NOTE: This API is similar to `task.create_task`. The only difference is that it provides several optional parameters instead of a consolidated `options` parameter.

Access

- Network members can use this API to create a task.
- If you have a hybrid network, guests can use this API to create a task for anyone in the network that they can see and that is named to them.

Method

POST

Parameters

`task_name` (required). The name of the task.

`assigned_members` (required). One or more user IDs. Separate multiple user IDs with commas (,).

`due_date` (optional). The date the task is due. Use the format YYYYMMDD. For example, 20110407 is for April 7, 2011.

`description` (optional). A description for the task.

`attachments` (optional). One or more attachment IDs. Separate multiple IDs with commas (,).

`action_type` (optional). The type of task. Options are:

- `none`. No type specified. This is the default value.
- `approval`. An approval task. The task assignee will need to approve this task. These tasks have slightly different states than other tasks (approved instead of completed).
- `consultation`. Use this for tasks where you are requesting consultation help.
- `directive`. Use this for tasks where you direct the assignee to do something.
- `invitation`. Use this for tasks where you invite the assignee to an event.
- `review`. Use this for tasks where you ask assignee to review something.

`flag` (optional). The importance of the task. Options are:

- `important`. The task importance flag turns orange when this value is set.
- `normal`. Default value.

`execute_mode` (optional). Whether or not all assigned members need to complete the task or just one member. Options are `all` or `only_one`.

`use_html` (optional). One of the following:

- `yes` indicates that HTML is allowed in the task description. This is the default behavior.
- `no` indicates that HTML is not allowed in the task description.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the ID of the new task.

task.update_task

Updates a task.

Access

- Network members can use this API to update their own tasks.
- If you have a hybrid network, guests can use this API to update a task that they created.

Method

POST

Parameters

`task_id` (required). The task ID.

`options` (optional). Specify one or more of the following options:

- `title`. The task name.
- `description`. The task description.
- `due_date`. The task due date. Use the format YYYYMMDD.
- `execute_mode`. Whether or not all assigned members need to complete the task or just one member. Options are `all` or `only_one`.
- `action_type`. The type of task. Options are:
 - `none`. No type specified. This is the default value.
 - `approval`. An approval task. The task assignee will need to approve this task. These tasks have slightly different states than other tasks (approved instead of completed).
 - `consultation`. Use this for tasks where you are requesting consultation help.
 - `directive`. Use this for tasks where you direct the assignee to do something.
 - `invitation`. Use this for tasks where you invite the assignee to an event.
 - `review`. Use this for tasks where you ask assignee to review something.
- `flag`. The importance of the task. Options are `important` or `normal` (the default).
- `use_html`. One of the following:
 - `yes` indicates that HTML is allowed in the task description. This is the default behavior.
 - `no` indicates that HTML is not allowed in the task description.

Separate the options with the pipe (`|`). For example: `options=title=Prepare annual report|due_date=20110228|action_type=review|flag=important`

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns `true`.

task.delete_task

Deletes a task.

Access

- Network members can use this API to delete their own tasks.
- If you have a hybrid network, guests can use this API to delete a task that they created.

Method

POST

Parameters

`task_id` (required). The task ID.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns `true`.

task.get_assigned_tasks

Gets the tasks assigned to yourself.

Access

- Network members can use this API to get the tasks that are assigned to them.
- If you have a hybrid network, guests can use this API to get the tasks that are assigned to them.

Method

GET

Parameters

`user_id` (required). Your user ID or login (email address). The `user_id` specified must match the caller ID.

`options` (optional). Specify one or more of the following options:

- `status`. The status of the tasks to get. Use one of the following:
 - `all`, for all tasks
 - `assigned`, for all assigned (open) tasks
 - `completed`, for all completed tasks
 - `declined`, for all declined tasks
- `from_due_date`. The earliest due date for the tasks to get. Use the format YYYYMMDD. For example, 20110407 is for April 7, 2011.
- `to_due_date`. The latest due date for the tasks to get. Use the format YYYYMMDD.
- `sort_by`. The order in which to sort the returned tasks. Options are:
 - `time_updated`. Sort by the time the tasks was last updated.
 - `time_created`. Sort by the time the tasks was created. This is the default setting.
 - `name`. Sort by the name of the task.
 - `duedate`. Sort by the task due date.

Precede the keyword with `+` to indicate an ascending sort order or `-` to indicate a descending sort order.

Separate the options with the pipe (`|`). For example:

```
options=status=all|sort_by=time_updated
```

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the following in an array. Each record includes:

guid	
owner_guid	
time_created	
time_updated	
title	
description	
status	
execute_mode	
due_date	
assignees	
action_type	
importance_flag	
attachment	This is an array of attachments
	guid
	name
	size
steps	This is an array of steps
	step_guid
	step_name
	execute_mode
	sequence
	due_date
	assignees
	status
	description

task.get_created_tasks

Gets the tasks created by a specific member.

Access

- Network members can use this API to get the tasks they created.
- If you have a hybrid network, guests can use this API to get the tasks they created.

Method

GET

Parameters

`user_id` (required). The user ID or login (email address).

`options` (optional). Specify one or more of the following options:

- `status`. The status of the tasks to get. Use one of the following:
 - `all`, for all tasks
 - `assigned`, for all assigned tasks (open, not completed)
 - `completed`, for all completed tasks
 - `declined`, for all declined tasks
- `from_due_date`. The earliest due date for the tasks to get. Use the format YYYYMMDD. For example, 20110407 is for April 7, 2011.
- `to_due_date`. The latest due date for the tasks to get. Use the format YYYYMMDD.
- `sort_by`. The order in which to sort the returned tasks. Options are:
 - `time_updated`. Sort by the time the tasks was last updated.
 - `time_created`. Sort by the time the tasks was created. This is the default setting.
 - `name`. Sort by the name of the task.
 - `duedate`. Sort by the task due date.

Precede the keyword with `+` to indicate an ascending sort order or `-` to indicate a descending sort order.

Separate the options with the pipe (`|`). For example:

```
options=status=all|sort_by=time_updated
```

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the following in an array. Each record includes:

guid	
owner_guid	
time_created	
time_updated	
title	
description	
status	
execute_mode	
due_date	
assignees	
action_type	
importance_flag	
attachment	This is an array of attachments
	guid
	name
	size
steps	This is an array of steps
	step_guid
	step_name
	execute_mode
	sequence
	due_date
	assignees
	status
	description

task.get_comments_of_task

Gets the comments for a specific task.

Access

- Network members can get comments on a task that they own or that is assigned to them.
- If you have a hybrid network, guests can use this API to get comments on a task they own or that is assigned to them.

Method

GET

Parameters

`task_id` (required). The task ID.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns an array of comments for the task.

task.add_comment_to_task

Adds a comment to a task.

Access

- Network members can add comments to a task that they own or that is assigned to them.
- If you have a hybrid network, guests can use this API to add comments to a task they own or that is assigned to them.

Method

POST

Parameters

`task_id` (required). The task ID.

`comment` (required). Comment to add to task.

`options` (optional). Specify the following option:

- `use_html`. One of the following:
 - `yes` indicates that HTML is allowed in the task comment. This is the default behavior.
 - `no` indicates that HTML is not allowed in the task comment.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the comment ID.

task.delete_comment_from_task

Deletes a comment from a task.

Access

- Network members can delete comments from a task that they own or that is assigned to them.
- If you have a hybrid network, guests can use this API to delete comments from a task they own or comments that they added to a task.

Method

POST

Parameters

`task_id` (required). The task ID.

`comment_id` (required). The ID of the comment to delete.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns `true`.

task.add_step

Adds a step to a task.

Access

- Network members can use this API to add steps to a task they own.
- If you have a hybrid network, guests can use this API to add steps to a task they own.

Method

POST

Parameters

`task_id` (required). The task ID.

`title` (required). The step title.

`assigned_members` (required). The members to which to assign the step. Specify the user IDs or logins (email addresses) for these members. Separate multiple members with commas.

`options` (optional). Specify one or more of the following options:

- `due_date`. The date that the step is due. Use the format YYYYMMDD. For example, 20110407 is for April 7, 2011.
- `description`. The step description.
- `execute_mode`. Options are:
 - `all`. All assignees must complete the step.
 - `only_one`. Only one step assignee must complete the step.
- `use_html`. One of the following:
 - `yes` indicates that HTML is allowed in the step description. This is the default behavior.
 - `no` indicates that HTML is not allowed in the step description.

Separate the options with the pipe (`|`). For example:

```
options=due_date=20120229|execute_mode=only_one
```

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the `step_ID` and `step_sequence` of the new step.

task.update_step

Updates a step within a task.

Access

- Network members can use this API to update steps within a task they own.
- If you have a hybrid network, guests can use this API to update steps within a task they own.

Method

POST

Parameters

`task_id` (required). The task ID.

`step_id` (required). The ID of the task step to update.

`options` (optional). Specify one or more of the following options:

- `title`. The step title.
- `due_date`. The date that the step is due. Use the format YYYYMMDD. For example, 20110407 is for April 7, 2011.
- `description`. The step description.
- `assigned_member`. The members to which to assign the step. Specify the user IDs or logins (email addresses) for these members.
- `execute_mode`. Options are:
 - `all`. All assignees must complete the step.
 - `only_one`. Only one step assignee must complete the step.
- `use_html`. One of the following:
 - `yes` indicates that HTML is allowed in the step description. This is the default behavior.
 - `no` indicates that HTML is not allowed in the step description.

Separate the options with the pipe (`|`). For example:

```
options=due_date=20120229|execute_mode=only_one
```

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the `step_ID` and `step_sequence` of the updated step.

task.delete_step

Deletes a step within a task.

Access

- Network members can use this API to delete steps within a task they own.
- If you have a hybrid network, guests can use this API to delete steps within a task they own.

Method

POST

Parameters

`task_id` (required). The task ID.

`step_id` (required). The step ID of the step to delete.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns `true`.

task.reorder_steps

Changes the order of the steps that belong to a task.

Access

- Network members can use this API to reorder steps within a task they own.
- If you have a hybrid network, guests can use this API to reorder steps within a task they own.

Method

POST

Parameters

`task_id` (required). The task ID.

`step_id` (required). The step ID of the step to reorder.

`direction` (required). The direction that the step should move. Options are:

- Up
- Down

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns `true`.

task.complete_task_assigned_to_me

Completes the task or task step assigned to you.

Note the following about this API:

- If the task has multiple steps, only the current step assigned to you is completed.
- You can only complete a step if it is your turn to complete the step. That is, if the person assigned the first step in a task has not completed it yet, and you are assigned the second step, you cannot use this method to complete your step.

Access

- Network members can use this API to complete tasks assigned to them.
- If you have a hybrid network, guests can use this API to complete tasks assigned to them.

Method

POST

Parameters

`task_guid` (required). The task ID.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the ID of the task.

task.accept_task_assigned_to_me

Accepts the task or task step assigned to you.

Note the following about this API:

- If the task has multiple steps, only the current step assigned to you is accepted.
- You can only accept a step if it is your turn to complete the step. That is, if the person assigned the first step in a task has not completed it yet, and you are assigned the second step, you cannot use this method to accept your step.

Access

- Network members can use this API to accept tasks assigned to them.
- If you have a hybrid network, guests can use this API to accept tasks assigned to them.

Method

POST

Parameters

`task_guid` (required). The task ID.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the ID of the task.

task.decline_task_assigned_to_me

Declines the task or task step assigned to you.

Note the following about this method:

- If the task has multiple steps, only the current step assigned to you is declined.
- You can only decline a step if it is your turn to complete the step. That is, if the person assigned the first step in a task has not completed it yet, and you are assigned the second step, you cannot use this method to decline your step.

Access

- Network members can use this API to decline tasks assigned to them.
- If you have a hybrid network, guests can use this API to decline tasks assigned to them.

Method

POST

Parameters

`task_guid` (required). The task ID.

`step_guid` (required). The step ID. Use 0 for the case of no steps created or for declining all steps.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the ID of the task.

task.acknowledge_task_notification

Accept acknowledgement that a task has been completed.

When someone completes a task that you assigned them, you receive a notification in your **Action Items** tab that they have completed the task. Each notification includes an **Accept** button that you need to click to remove the notification from your **Action Items** tab. By using this API, you can remove these notifications from your **Action Items** tab.

Access

- Network members can use this API to acknowledge tasks assigned to them.
- If you have a hybrid network, guests can use this API to acknowledge tasks assigned to them.

Method

POST

Parameters

`task_guid` (required). The task ID.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the ID of the task.

task.get_step_assignees

Get the assignees for a task step.

Access

- Only the assignees of the task and the task owner can see the assignees of the step.
- If you have a hybrid network, guests can use this API to get step assignees for tasks they own or tasks that are assigned to them.

Method

GET

Parameters

`task_id` (required). The task ID.

`step_id` (required). The step ID.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the following in an array. Each record includes:

<code>guid</code>
<code>name</code>
<code>status</code>
<code>time_updated</code>

task.get_task_by_id

Get task by ID.

Access

- Only the assignees of the task and the task owner can see the task.
- If you have a hybrid network, guests can use this API to get tasks they own or tasks that are assigned to them.

Method

GET

Parameters

`task_id` (required). The task ID.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns the following in an array. Each record includes:

guid	
owner_guid	
time_created	
time_updated	
title	
description	
status	
execute_mode	
due_date	
assignees	
attachments	
This is an array of attachments	
guid	
size	
name	
steps	
This is an array of steps	
step_guid	
step_name	
execute_mode	
sequence	
due_date	
description	
status	

Chapter 2 APIs
Using the API

assignees
flag
action_type

task.add_attachments

Add attachments to a task.

Access

- Only the assignees of the task and the task owner can add attachments to a task.
- If you have a hybrid network, guests can use this API to add attachments for tasks they own or tasks that are assigned to them.

Method

POST

Parameters

`task_id` (required). The task ID.

`attachment_ids` (required). The attachment ID. For multiple attachments, separate the attachment IDs with commas.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns `true`.

task.remove_attachments

Remove attachments from a task.

Access

- Only the assignees of the task and the task owner can remove attachments from a task.
- If you have a hybrid network, guests can use this API to remove attachments from tasks they own or tasks that are assigned to them.

Method

POST

Parameters

`task_id` (required). The task ID.

`attachment_ids` (required). The attachment ID. For multiple attachments, separate the attachment IDs with commas.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns `true`.

task.add_assignees

Add assignees to a task.

Access

- Only the assignees of the task and the task owner can add more assignees to a task.
- If you have a hybrid network, guests can use this API to add assignees to tasks they own or tasks that are assigned to them.

Method

POST

Parameters

`task_id` (required). The task ID.

`assignees` (required). The user IDs for the new task assignees. Separate multiple user IDs with commas.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns `true`.

task.remove_assignees

Remove assignees from a task.

Access

- Only the assignees of the task and the task owner can remove assignees from a task.
- If you have a hybrid network, guests can use this API to remove assignees from tasks they own or tasks that are assigned to them.

Method

POST

Parameters

`task_id` (required). The task ID.

`assignees` (required). The user IDs for the task assignees to remove. Separate multiple user IDs with commas.

Status Code

Code	Description
0	Success. Any other value indicates that the method failed.
-1	Failed.
-20	Failed due to invalid or expired token.

Returns

On success, this method returns `true`.

3

Customizing Clearvale

This chapter contains the following topics:

- “Signing Into Clearvale Through Another Application,” next

Signing Into Clearvale Through Another Application

If you are using Clearvale with another application, you may want to use that application to authenticate (sign in) Clearvale users. By doing this:

- Your application login page appears instead of the Clearvale login pages when users try to log into their Clearvale network.
- Your application login page handles signing users into your application and Clearvale.
- If your application includes a link to a Clearvale network, the user does not need to additionally sign into Clearvale—they are already signed in.

Configuring Your Network to Use a Different Login Page

To use a custom login page, contact BroadVision about provisioning the Login Page Redirection feature for your account. Once provisioned, use the Clearvale Management Center to configure your network to use a custom login page.

To configure your network to use a different login page

1. Sign into the Clearvale Management Center as an Account Administrator.
2. Navigate to your account page and click the **Manage** link next to the network for which you want to enable this feature.
3. Scroll down to the Clearvale Services area and click **Enable** next to Clearvale API and then click **Enable** again next to Login Page Redirection.

You need both of these services enabled for your network to use an external login page.

As a result of enabling these features, a Login Page Redirection block appears.

4. Click **Edit** in the block to specify the URL for the login page to use for your network.
5. Click **Save**.

For information on using the Clearvale Management Center to manage your Clearvale account and networks, see the *Clearvale Account Manager Guide*.

Editing Your Login Page to Authenticate for Clearvale

You probably already have a login page that you want to use for your network. In this case, edit the page so that it authenticates for your Clearvale network as well as your application.

To edit a login page to authenticate for Clearvale

1. Open your login page in the appropriate editor.
Typically these might be .jsp or .php pages.
2. Within your login page, use the following Clearvale APIs to sign a user into Clearvale:
 - First, using a Clearvale user name for someone with network administrator privileges, get an authentication token by calling `cvauth.authenticate`.
 - Next, you can get the authentication token for other users by calling `cvauth.get_token`. This method requires that you pass the `auth_token` returned by `cvauth.authenticate`.
 - Set the cookie for the user by calling the `cvauth.set_cookie`. This establishes the session for the user.
 - If a user registers or is invited, use `membership.create_user` and `membership.add_member` to add members to the network.
3. Save your changes

Which Clearvale Login Pages Are Affected?

Clearvale lets you specify your own login page to use instead of the standard login pages that Clearvale provides. The standard login pages are as follows:

Page	Path
Network login page	<i>network_url/login.php</i>
Login page that appears when you click Lost Password on the login page	<i>network_url/account/forgotten_password.php</i>
Network registration page	<i>network_url/account/register.php</i>
Network invitation page	<i>network_url/mod/invitemass/joinwizard.php</i>
Ecosystem login page	<i>ecosystem_URL/ecosystem/member-eco-login.php</i>
Ecosystem registration page	<i>ecosystem_URL/ecosystem/register.php</i>

If you specify a different login page to use, your login page appears instead of the above pages.

Troubleshooting

If you run into problems with your custom login page, and cannot log into Clearvale, the Clearvale Account Manager can sign into the Clearvale Management Center and disable the Login Page Redirection service for the network. By disabling this, the custom login page is no longer available and the standard Clearvale login pages are used.

4

External Widgets

This chapter covers creating external widgets and then installing them for your network. It also covers sharing your external widgets with other networks.

This chapter assumes that you have the Owner or Administrator role on your network.

What Are External Widgets?

An external widget is a small application that is installed as a widget within a network and offers dynamic content from an external application.

Once you create and install an external widget, it operates like any other widget. Slight differences include:

- External widgets appear in a separate section of the widget gallery, labeled Global Library.
- External widgets require that you edit them after placing them on the page. By default, the widget is offline. You need to set it online and agree to privacy terms as well as set any required widget parameters.

After you create an external widget, you can choose to let other networks use your external widget. You have two choices for sharing external widgets:

- Let other networks within your Clearvale account use your external widget. To do this, install your external widget and then add it to your account library. (See [“Adding Your Widget to the Account Library” on page 232.](#))
- Let other networks within your Clearvale installation use your external widget. To do this, install your widget and then add it to the global library. (See [“Adding Your Widget to the Global Library” on page 233.](#)) All networks within your Clearvale installation can access your widget. When you upload a widget to the global library, BroadVision must approve the widget before others can see or access the widget.

Creating an External Widget

To create an external widget

1. Make sure that your network is part of your Clearvale account in the Clearvale Management Center.
See the *Clearvale Account Manager Guide* for information on managing your Clearvale account.
2. Make sure that your Clearvale network has the appropriate services enabled for it.

Service	Description
Install External Widgets	Required. This service adds the Admin>External Network tab for network administrators. It allows the network administrator to install external widgets on the network and to share them with the account widget library.
External Widget Library	Optional. This service is only needed if you want to share your widget with the global widget library.
Clearvale API	Optional. This service is only needed if your external widget application makes use of the Clearvale APIs.

Again, see the *Clearvale Account Manager Guide* for information on managing the services enabled for your networks.

3. If you have not done this yet, download a provider token from BroadVision.
See “[Downloading a Provider Token From BroadVision](#)” on page 219.
4. Create a web application for the external widget.
See “[Creating a Web Application for Your External Widget](#)” on page 220.
5. Create an XML manifest file that defines the widget.
See “[Creating an External Widget Manifest File](#)” on page 230.

Downloading a Provider Token From BroadVision

The provider token allows you to authenticate the request message and encrypt data when communicating with Clearvale. You need to include this token in all external widget web applications. You need only one provider token per Clearvale account.

To download your provider token

1. Sign into your Clearvale network as an administrator.
2. Select **Admin>External Widgets>Manage Widgets**.
3. Click the **Download Provider Token** link.

A file is downloaded that contains the token. You will need this same token in any additional external widgets that you create. Note the following about the token:

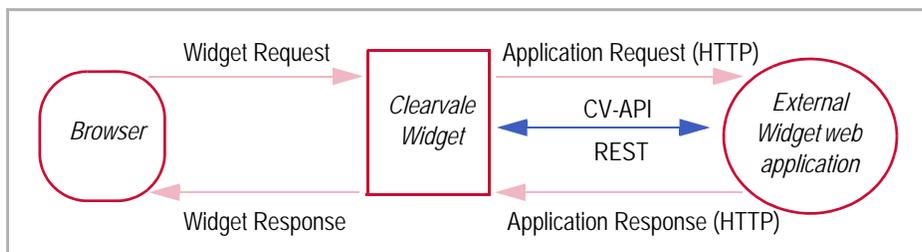
- If you forget your provider token, you can download another copy by clicking the **Download Provider Token** link.
- If your provider token is compromised in some fashion, and you want to use a different token, you can regenerate the token by using the **Regenerate Provider Token** link. However, if you regenerate your token, your old token is invalid and you need to edit all of your external applications that serve the widget to use the new token.

Creating a Web Application for Your External Widget

You need to create a small web application that will display your data in the Clearvale network. Keep the following guidelines in mind when developing the application:

- Your application must have a URL. Clearvale calls the application’s URL to render the widget. Use the `widgetAction` parameters to implement the actions that the widget needs to take. See “Authentication,” next.
- Use the Clearvale API for calls between the external server and Clearvale.

The following diagram shows how the external widget acts as an interface between the browser and the external application. The browser does not directly call the external application.



Authentication

There are two types of authentication that pertain when using Clearvale external widgets:

- **Clearvale API authentication.** In this case, the widget inherits the Clearvale authentication token from the logged in user. This token needs to be decrypted. You need to use the Cipher class for the decrypting. You can find sample code for this class in the “[PHP Example with Clearvale API Authentication](#)” on page 228.
- **Widget authentication.** In this case, Clearvale authenticates with the external widget server. To use this version of authentication, use `widgetAction=authenticationRequest`, which is described below.

widgetAction Parameters

Your application needs to be able to handle the data that `widgetAction` passes. Possible values for `widgetAction` are:

- `widgetAction=authenticationRequest`
- `widgetAction=widgetParams`
- `widgetAction=widgetBody`

Each of these are described below.

authenticationRequest

The `widgetAction=authenticationRequest` message is sent by Clearvale to authenticate the external content widget server. The external content widget server must respond with the following data in the body:

```
widgetAction=authenticationResponse
&widgetName='Name of the widget'
&widgetTOTPToken='token'
```

where *token* is the token generated by the HMAC_TOTP function. HMAC_TOTP is a function needed to build the HMAC TOTP implementation. For more information about HMAC_TOTP, see <http://tools.ietf.org/html/draft-mraihi-totp-timebased-07>.

Pseudocode for this might look as follows:

```
IF action equal 'authenticationRequest' THEN
    // Need to respond to Authentication request with password generated
    // with HMAC TOTP algorithm

    data = provider_token + 'authenticationResponse' ;

    password = HMAC_TOTP (data ) ;

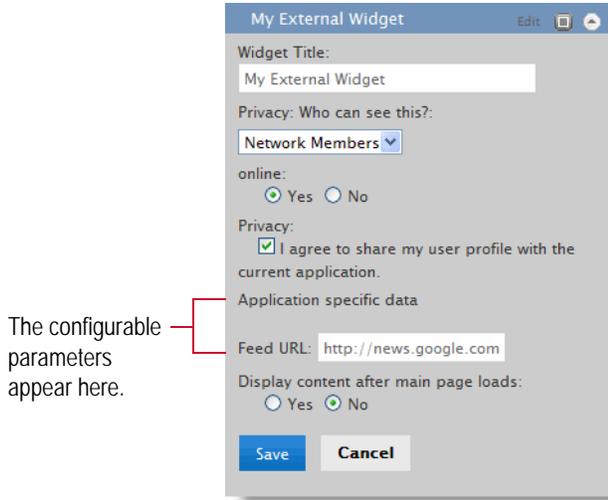
    send_output_response ('widgetAction=authenticationResponse') ;

    send_output_response ('widgetName=RSS');

    send_output_response ('widgetTOTPToken='+ password );
```

widgetParams

Use `widgetAction=widgetParams` for user-configurable parameters. Any parameters that you define appear when you click **Edit** in the upper-left corner of an external widget in your application. For example, if your external widget requires a feed URL, it might look as follows:



When the network member configures the widget, and saves the changes, the field values are saved in Clearvale. When they access the page again, Clearvale sends the saved parameter values along with the `widgetAction=widgetBody` parameter to the external widget. The external widget gets the needed parameters, processes the request, and then returns the appropriate data back to Clearvale.

widgetBody

Use `widgetAction=widgetBody` to define the body of the external widget to display at runtime.

Make sure that your application can handle any requests that are sent to it. If all of the required parameters are sent, your application should process the request and return the appropriate HTML to display in the widget. Your application has complete control to display any HTML data.

Examples

Pseudocode
Example

The following pseudocode shows how to create an RSS external widget. Keywords are shown in 'single quotes':

```
// Load all the common data

action = get_parameter_from_request ( 'widgetAction' ) ;
provider_token = Load_Provider_Token ( ) ;

IF action equal 'authenticationRequest' THEN
    // Need to response for Authentication request with password generating
    // with HMAC TOTP algorithm

    data = provider_token + 'authenticationResponse' ;
    password = HMAC_TOTP (data ) ;
    send_output_response ( 'widgetAction=authenticationResponse' ) ;
    send_output_response ( 'widgetName=RSS' ) ;
    send_output_response ( 'widgetTOTPToken='+ password ) ;

ENDIF

IF action equal 'widgetParams' THEN
// Building an HTML form segment when ECW configuration is edited by user

    param = '<div id="widgetParams"> <p>' ;
    param = param + 'Feed URL: <input type="text" name = "params[feedUrl]"
size="80" ' ;
    param = param + 'value= "http://somefeed.com/feed"/>' ;
    param = param + '</p></div>' ;

    send_output_response (param) ;

ENDIF

IF action equal 'WidgetBody' THEN

    cvapiencrypted_token = get_parameter_from_request('widgetCVAPIToken') ;
    cvtoken = DECRYPT('blowfish', provider_token,
        base64_decode(cvapiencrypted_token), MODE_ECB ) ;

    do_some_cv_api_call (cvtoken) ;

    display_mode= $get_parameter_from_request('widgetDisplayState');

    IF ( display_mode equal 'Normal' ) THEN
        data = prepare_data_for_widget_display() ;
    ELSE IF display_mode equal 'Expanded' ) THEN
        data = prepare_data_for_full_view() ;
    ENDIF

    send_output_response (data) ;

ENDIF
```

PHP Example The following PHP example shows how to create an RSS feeder external widget. This example uses `HMAC_TOTP` along with the sliding window protocol to securely manage the data transmission.

```
<?php
// ***** MAIN program
// Load the $provider_token variable
include('providertoken.php');

// Main action handler
$action = ($_REQUEST['widgetAction'])?$_REQUEST['widgetAction']:false;
$token =
($_REQUEST['widgetTOTPToken'])?$_REQUEST['widgetTOTPToken']:false;

error_log("ECW-RSS Widget Server:$provider_token:: " . var_export($_REQUEST,
true));
switch ($action){
    case 'widgetParams':
        // This code adds the HTML input tag on the widget configuration page
        $msg = ($_REQUEST['message'])?$_REQUEST['message']:'The world';

        $param = '<div id="widgetParams">' ;
        $param .= "<p>Message: " ;
        $param .= "<input type='text' name = 'params[message]' size='10'
value= '$msg'/>" ;
        $param .= '</p></div>' ;

        // Send the response back
        echo $param ;
        break;

    case 'authenticationRequest' :

        $algorithm = $_REQUEST['securityParam']['algol'];
        $slidingwindow = $_REQUEST['securityParam']['slidingWindow'];

        $config = new stdClass();
        $config->algol = $algorithm ;
        $config->slidingWindow = $w ;
        $config->passwordLength = strlen($token);

        $password = HMAC_TOTP($provider_token.'authenticationResponse',
$config ) ; //authenticationResponse
        $resp = 'widgetAction=authenticationResponse' ;
        $resp .= '&widgetName=RSS' ;
        $resp .= "&widgetSessionLength=1" ; // 5 hours
        $resp .= "&widgetTOTPToken=$password" ;
        $resp .= "&securityParam[slidingWindow]=10" ;
        $resp .= "&securityParam[passwordLength]=7" ;

        echo $resp ;
        break;

    case 'WidgetBody':
    default:
```

```

        $cvapi_encr =
($REQUEST['widgetCVAPIToken'])?$_REQUEST['widgetCVAPIToken']:false;
        if ( !empty($cvapitoken_encr) ){
            $cvapitoken =trim(mcrypt_decrypt('blowfish', $provider_token,
base64_decode($cvapi_encr), MCRYPT_MODE_ECB, null));
        }
        // Now CV-API can be called without the need to use username & password.

// prepare the output to be display
$msg = ($_REQUEST['message'])?$_REQUEST['message']:'';

$display= $_REQUEST['widgetDisplayState'] ;
if ( $display == 'Normal' ){
    echo '<div><i>This is example of ECW server</i>' ;
    echo '<BR/>' ;
    echo $msg ;
    echo '</div>' ;
}
if ( $display == 'Expanded' ){
    echo '<div><H1><B>Welcome</B></H1>' ;
    echo '<BR/>' ;
    echo '<BR/>This is example of ECW server' ;
    echo '<BR/>As you can see this has plenty data' ;
    echo '<BR/>Message for you: ' ;
    echo '<BR/>' . $msg ;
    echo '</div>' ;
}
break;
}

// ***** MAIN program END

// HMAC_TOTP PHP implementation

function HMAC_TOTP ( $key, $config ){

    // Parameter for TOTP.
    // $config->timeStamp ; // use specific time stamp
    // $config->algor ; // Algorithm to use, default sha1
    // $config->slidingWindow ; // W to use, default 30 seconds
    // $config->passwordLength ; // leng to use, default 9, false will not
    // truncate the hash result.

    if ( empty($config->timeStamp) )
        $timestamp = time();
    else
        $timestamp = $config->timeStamp ;
    if ( empty($config->algor) )
        $algo = DEFAULT_ALGORITHM;
    else
        $algo = $config->algor;

    if ( empty($config->slidingWindow) )
        $W = DEFAULT_SLIDING_WINDOW ;

```

```
        else
            $W = $config->slidingWindow ;

        if ( empty($config->passwordLength) )
            $len = 9 ;
        else
            $len = $config->passwordLength ;
        $timeslice = (int)($timestamp/$W) ;
        if ( $timeslice < 0 )
            $timeslice = 1 ;
        $counter = $timeslice ;

        $hash = oath_hotp ($key, $counter, $algo ) ;

        if ( $len ){
            $hash = oath_truncate( $hash, $len ) ;
        }
        return $hash ;
    }

function oath_hotp ($key, $input_counter, $algo=DEFAULT_ALGORITHM){
    $counter = $input_counter ;
    //the counter value can be more than one byte long, so we need to go
multiple times
    $cur_counter = array(0,0,0,0,0,0,0,0);
    for($i=7;$i>=0;$i--){
        {
            $cur_counter[$i] = pack ('C*', $counter);
            $counter = $counter >> 8;
        }
    }
    $bin_counter = implode($cur_counter);
    // Pad to 8 chars
    if (strlen ($bin_counter) < 8)
    {
        $bin_counter = str_repeat (chr(0), 8 - strlen ($bin_counter)) .
$bin_counter;
    }
    // HMAC
    $hash = hash_hmac ($algo, $bin_counter, $key);
    return $hash;
}

function oath_truncate($hash, $length = 9){
    if ( $length > 9 ) $length=9 ; // maximum 9 digits: pow= 1 000 000 000
    if ( $length < 6 ) $length=6 ; // minimum 6 digits: 1 000 000 000
    // Convert to dec
    foreach(str_split($hash,2) as $hex)
    {
        {
            $hmac_result[]=hexdec($hex);
        }
    }
    // Find offset
    $offset = $hmac_result[19] & 0xf;
    // Algorithm from RFC
```

```
$pow = pow(10,$length) ;
$res =
(
(($hmac_result[$offset+0] & 0x7f) << 24 ) |
(($hmac_result[$offset+1] & 0xff) << 16 ) |
(($hmac_result[$offset+2] & 0xff) << 8 ) |
($hmac_result[$offset+3] & 0xff)
) % $pow ;

while ( strlen ($res) < $length )
    $res = '0'.$res ;

return $res;
}
```

The `providertoken.php` file looks as follows:

```
<?php
$provider_token='niNfZeVzukftslget/gRA==' ;
?>
```

Chapter 4 External Widgets

Creating an External Widget

PHP Example with Clearvale API Authentication The following PHP example shows how to create an RSS feeder external widget. This example uses Clearvale API authentication to get the user profile data. It includes the Cipher class for decrypting the Clearvale API authentication token.

```
<?

$action = $_REQUEST["widgetAction"];
if ( $action == 'widgetParams' ) {
    // if we're in configuration mode
    $url = $_REQUEST['feedUrl'];
    if ( empty($url) )
        $url = 'http://news.google.com/news?ned=us&topic=h&output=rss' ;
    $size = strlen($url) ;
    $param = '<div id="widgetParams">' ;
    $param .= "<p>Feed URL: <input type='text' name = 'params[feedUrl]'
size='20' " ;
    $param .= 'value= "' . $url . "'></p>' ;
    $param .= '</div>' ;
    echo $param ;
} else {
    // if we're in normal display mode

    $url = $_REQUEST['feedUrl'];
    if (!empty($url) ) {
        // sample provider token--replace with your own
        $provider_token = '8eHdg5D7LWXfxn0Y9yoTPg==' ;
        // decrypt the CV API token
        $cvapidata = $_REQUEST['widgetCVAPIToken'] ;
        if ( !empty($cvapidata)){
            $sec = new Cipher (base64_decode($provider_token));
            $cvapitoken = $sec->decrypt($cvapidata);
        }
        // call CV to get visitor's profile
        $profileurl =
"http://".$_REQUEST["widgetNetworkName"].".clearvale.com/pg/cvapi/rest/xml?me
```

```

thod=profile.get_profile_for_user&auth_token=$cvapitoken&user_id=".$_REQUEST[
"widgetLoginId"];
    $profile = simplexml_load_file($profileurl);
    $name = $profile->result->array_item->name;
    print "<h1>Here is the news for $name</h1>";
    $contents = simplexml_load_file($url);
    foreach ($contents->channel->item as $i) {
        print "<li><a href='$i->link'>$i->title</a></li>";
    }
}
/* print "</ul><br/><br/><h1>Here are the request parameters</h1><ul>";
foreach ($_REQUEST as $k=>$v) {
    print "<li>$k = $v</li>";
}
print "</ul>";*/
}

// used in CV API token decryption
// requires mcrypt module to be enabled

class Cipher {

    private $securekey, $iv, $encryptAlgol;
    function __construct( $textkey, $encryptalgol =
MCRYPT_BLOWFISH, $hash_f = 'sha256' ) {
        $this->securekey = $textkey;
        $this->iv = mcrypt_create_iv(32, MCRYPT_RAND);
        $this->encryptAlgol = $encryptalgol ;
    }

    function encrypt($input) {
        return base64_encode(mcrypt_encrypt($this->encryptAlgol, $this-
>securekey, $input, MCRYPT_MODE_ECB, $this->iv));
    }

    function decrypt($input) {
        return trim(mcrypt_decrypt($this->encryptAlgol, $this->securekey,
base64_decode($input), MCRYPT_MODE_ECB, $this->iv));
    }
}
?>

```

More Examples For additional examples, check the Developer Forum community on [Clearvale Connect](#).

Creating an External Widget Manifest File

Each widget requires a manifest file that defines the external widget for Clearvale. The manifest specifies the name of the widget, its URL, and the type of page on which the widget can appear (for example, the network page, a community page, or a personal page).

To create a manifest file

1. Open a file in a text editor.
2. Add the following to define the external widget for Clearvale:
 - The `<name>` parameter identifies the widget.
 - The `<url>` parameter identifies the URL for the widget.
 - The `<location>` parameter identifies the types of pages on which the widget can appear. Options for this parameter are:
 - The keyword `all` to allow the widget on all types of page. For example:
`<location>all</location>`
 - Or, one or more of the following keywords:
`network` (allow the widget on the Network page),
`community` (allow the widget on any community page)
`personal` (allow the widget on any personal page)
For example: `<location>network</location>`
To allow a widget on two or more types of pages, separate the options with a comma. For example: `<location>network,personal</location>`
 - The `<private>` parameter indicates who can see the widget when it appears on a personal page. To allow everyone to see the widget, use `<private>0</private>`. If you omit the parameter, or set it to `1`, just the page owner can see the widget.
3. Save the file with an `.xml` extension.

Sample Manifest File

```
<widget>
  <name>Acme Widget</name>
  <url>http://sp.com/acme-widget.do</url>
  <location>all</location>
</widget>
```

Installing Your External Widget

Once you have created your external widget and its XML manifest file, you can install it for use on your network. You must have the owner or administrator role on your network to install an external widget.

To install an external widget

1. Sign into your Clearvale network as an administrator.
2. Select **Admin>External Widgets>Manage Widgets**.
3. Scroll down to the Install Widget area and click **Browse** to navigate to the manifest file for your external widget.
4. Click **Install**.

At this point, your widget is available for use on your network and is part of the local widget library. It automatically appears on the page layout editor for the page types specified in the manifest file. For example, if you selected `network` as the location for the widget, click the **Network** tab for your network, and then click the **Edit Layout** tab. In the Widget Gallery on the right-side of the page, scroll down to see your external widget.

Adding Your Widget to the Account Library

You can share your external widget with other Clearvale networks in your Clearvale account. To do this, add your external widget to your account library. Any widget added to your account library is available for installation by other networks in the same Clearvale account.

To add your widget to the account library

1. Sign into your Clearvale network as an administrator.
2. Select **Admin>External Widgets>Manage Widgets**.
3. Scroll down to the Installed Widgets area and click **Submit to Library** next to the external widget to submit to the account library.
4. Click **Account library** as the library in which to publish your widget.
5. Enter a name for the widget.
By default, the name you specified when you installed the external widget is used.
6. Optionally, specify the following for the external widget:
 - A URL for the logo to use for the widget. The logo appears next to the widget on the Account Library page.
 - An overview for the widget. The overview also appears next to the widget on the Account Library page.
 - Details for the widget. The details are visible when you display the details page for the particular external widget.
7. Click **Submit for Approval**.

At this point, your widget is added to your account library. For information on using an external widget in another network in your account, see [“Installing a Widget from the Account or Global Library into Your Network”](#) on page 235.

Adding Your Widget to the Global Library

Alternatively, you can also share your external widget with all other Clearvale networks on your Clearvale installation. In this case, when you submit the widget for others to use, BroadVision must first approve the widget before others have access to it.

To submit a widget for approval to the global library

1. Sign into your Clearvale network as an administrator.
2. Select **Admin>External Widgets>Manage Widgets**.
3. Scroll down to the Installed Widgets area and click **Submit to Library** next to the external widget to submit to the account library.
4. Click **Global library** as the library in which to publish your widget.
5. Enter a name for the widget.
By default, the name you specified when you installed the external widget is used.
6. Optionally, specify the following for the external widget:
 - A URL for the logo to use for the widget. The logo appears next to the widget on the Global Library page.
 - An overview for the widget. The overview also appears next to the widget on the Global Library page.
 - Details for the widget. The details are visible when you display the details page for the particular external widget.
7. Click **Submit for Approval**.

At this point, your widget is waiting for approval to be added to the global library. See [“Approving Global Library Widgets,”](#) next.

Approving Global Library Widgets

Global widgets are in a state of Pending Approval until an Operations Manager or Support Manager approves or rejects the widget by using the Clearvale Management Center:

- If approved, the widget's status changes to Approved. The external widget is added to the Global Library of widgets. This means that any Clearvale network in your installation can use this widget. If you select **Admin>External Widgets>Global Library**, you will see the external widget. You will also receive an email notification when your widget was approved.
- If rejected, the status changes to Rejected. You can continue using the widget on your network, but it is not added to the Global Library. You will also receive an email notification when your widget was rejected.

Installing a Widget from the Account or Global Library into Your Network

You can also install into your network any external widgets listed in the global library. External widgets are only added to the global library after someone submits them for approval and BroadVision approves them.

To install an account widget into your network

1. Sign into your Clearvale network as an administrator.
2. Select **Admin>External Widgets>Account Library**.
3. Click the **Install** link next to the widget to install into your network.

This widget then appears in your network, on the page layout editor. Note that the widget may not appear in all page layout editors, it depends on the types of pages for which the widget was designed and configured (specified by the `<location>` parameter in the XML manifest file).

To install a global widget into your network

1. Sign into your Clearvale network as an administrator.
2. Select **Admin>External Widgets>Global Library**.
3. Click the **Install** link next to the widget to install into your network.

This widget then appears in your network, on the page layout editor. Note that the widget may not appear in all page layout editors, it depends on the types of pages for which the widget was designed and configured (specified by the `<location>` parameter in the XML manifest file).

Configuring an External Widget

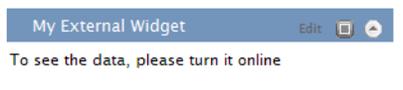
Just like regular widgets that you place on a page, the external widgets are also dragged and dropped onto the page layout. Then, on the page itself, you can configure the widget so it contains the right information.

To configure an external widget

1. Place external widgets on the pages in your network where you want them.

To place an external widget on a page, click the **Edit Layout** tab on the page where you want to place the widget. Drag the widget from the Global Library area of the Widget Gallery and drop it on the page layout where you want it to appear. Save your page layout changes.

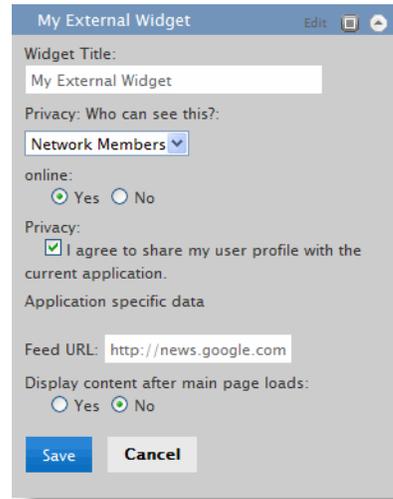
2. Click the **Edit** link for the external widget to set any preferences.



3. Enter values for the following standard settings for all external widgets:
 - Set the **Widget Title**. The title should describe the external widget.
 - Set the **Privacy: Who can see this?** setting for the widget. Select a value such as Network Members or Community Members.
 - Set the **Online** button to **Yes** to enable the external widget. By default, it is set to **No**, and the external widget is offline.
 - Check the **Privacy** checkbox if your application requires some profile-related data retrieved from Clearvale.
 - **Display content after main page loads**. This setting typically determines how the page displays content. However, external widgets are always displayed after the main page loads. This setting instead controls whether the external widget uses an iFrame or not. **No** is the default value and assumes that you are not using an iFrame. Set it to **Yes** if you are using iFrame.

4. Enter any external widget-specific data in the **Application specific data** area.

For example:



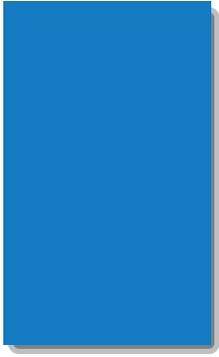
The screenshot shows a configuration dialog box titled "My External Widget". It contains the following fields and options:

- Widget Title:** A text input field containing "My External Widget".
- Privacy: Who can see this?:** A dropdown menu set to "Network Members".
- online:** Radio buttons for "Yes" (selected) and "No".
- Privacy:** A checked checkbox for "I agree to share my user profile with the current application."
- Application specific data:** A section header.
- Feed URL:** A text input field containing "http://news.google.com".
- Display content after main page loads:** Radio buttons for "Yes" and "No" (selected).
- Buttons:** "Save" and "Cancel".

5. Click **Save**.

Removing External Widgets

- To remove a particular instance of an external widget from a page in your network, click the **Edit Layout** tab for the page where the external widget appears, and then drag the external widget back into the widget gallery. Save your changes. The widget no longer appears on the page.
- To remove an external widget from the widget gallery in your network, as an administrator, select **Admin>External Widgets>Manage Widgets**. Then, under Installed Widgets, click **Uninstall** next to the external widget to uninstall.
- To remove an external widget from the account library or global library, select **Admin>External Widgets>Manage Widgets**. Then, under Submitted Widgets, click **Remove** next to the external widget to remove. If you only want to remove the external widget from the library temporarily, use the **Remove from Library** option instead. When you use this option, a link appears to easily re-add the external widget to the library.



Index

A

- about this book 5
- activity API 17
- activity points, getting 33
- admin authentication 80
- analytics API 33
- APIs 13
- auth_token parameter 14
- authentication API 78
- authenticationRequest 221

B

- blogs API 34
- book organization 5
- bookmarks API 53

C

- calendar API 59
- communities API 67
- cvauth API 78

D

- documents API 85

E

- encoding request parameters 14
- events API 59
- external widget
 - installing 231

- external widgets
 - about 217
 - adding to account library 232
 - adding to global library 233
 - configuring 236
 - creating 218
 - creating web application for 220
 - creating widget manifest file 230
 - removing 238
 - sample code 223

F

- features overview 8
- files API 85
- forums API 119

G

- glossary of common terms 11
- groups API 130

H

- HTTPS protocol 79, 132, 134

J

- jsoncallback parameter 14

L

- logging into Clearvale through another application 213

Index

login page
 changing 213
 redirection 214

M

membership API 130
method parameter 14
microblog API 147

O

org API 239

P

pages API 156
polls API 164
POST request syntax 14
profile API 171
provider token 219

R

request format 14
request parameters, encoding 14
response format 15
result response 15

S

search API 179
signing into Clearvale through another application 213
statistics API 181
status response 15

T

tasks API 184
terms used in clearvale 11
typographical conventions 6

U

URL encoding 14
user profile API 171
users API 130

W

widgetAction parameters 221
widgetBody 222
widgetParams 222
wiki pages API 156