Fumaggo

eZ Tube extension

Documentation
version 1.0

Copyright and trademarks
Copyright © 2011 Fumaggo Internet Solutions, the Netherlands. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Other product and company names mentioned in this manual may be the trademarks of their respective owners. We use trademark names in an editorial fashion to the benefit of the trademark holder; therefore, these names are not marked with trademark symbols. All terms known to be trademarks have been appropriately capitalized. We cannot attest to the accuracy of this usage, and usage of a term in this book should not be regarded as affecting the validity of any trademark or servicemark.

Table of Contents

1.   Requirements

The following requirements exists for using eZ Tube extension:

- o eZ Publish version: eZ Publish 4.5 or higher.

- o PHP version: 5.2.1 and higher

2.   Installation

1. Unpack/unzip

Unpack the zip package into the 'extension' directory of your
eZ Publish installation.

2. Activate extension

Activate the extension by using the admin interface (Setup -> Extensions) or by
prepending eztube to ActiveExtensions[] in settings/override/site.ini.append.php:

```
[ExtensionSettings]
ActiveExtensions[]=eztube
```

3. Regenerate autoload array

Check the eZ Publish docs for your version on how this is done, or go to Setup →
Extensions and click on the button there.

4. Rewrite rules for eztube extension

Add the following rewrite rule:

```
RewriteRule ^/var/[^/]+/cache/eztube/.* - [L]
```

5. Add permission for the users

*Add permissions so visitors of your site can view videos, comments, profiles, etc. Go to
Setup → User Accounts → Roles and Policies, and select a role to edit, for example
"Anonymous". Click on the "New Policy" button and select the module "eztube". Grant
access to all functions, or add the required functions to the role one by one. Save the
modified role after you are done adding the eZ Tube policies. Check the eZ Publish docs
for more details on modifying roles and policies.*

6. Clear caches

Clear INI and template caches (from admin 'Setup' tab or commandline).

3.   Introduction

3.1   Homepage

After you have completed the installation and configuration of your site (see section 4, "Getting Started"), you can access the video homepage at http://[sitename]/[site_access]/eztube/home. By default, the homepage consists of five rows, displaying the search results of configurable YouTube queries. See section 4.2 for more information on how to change the contents of the homepage.
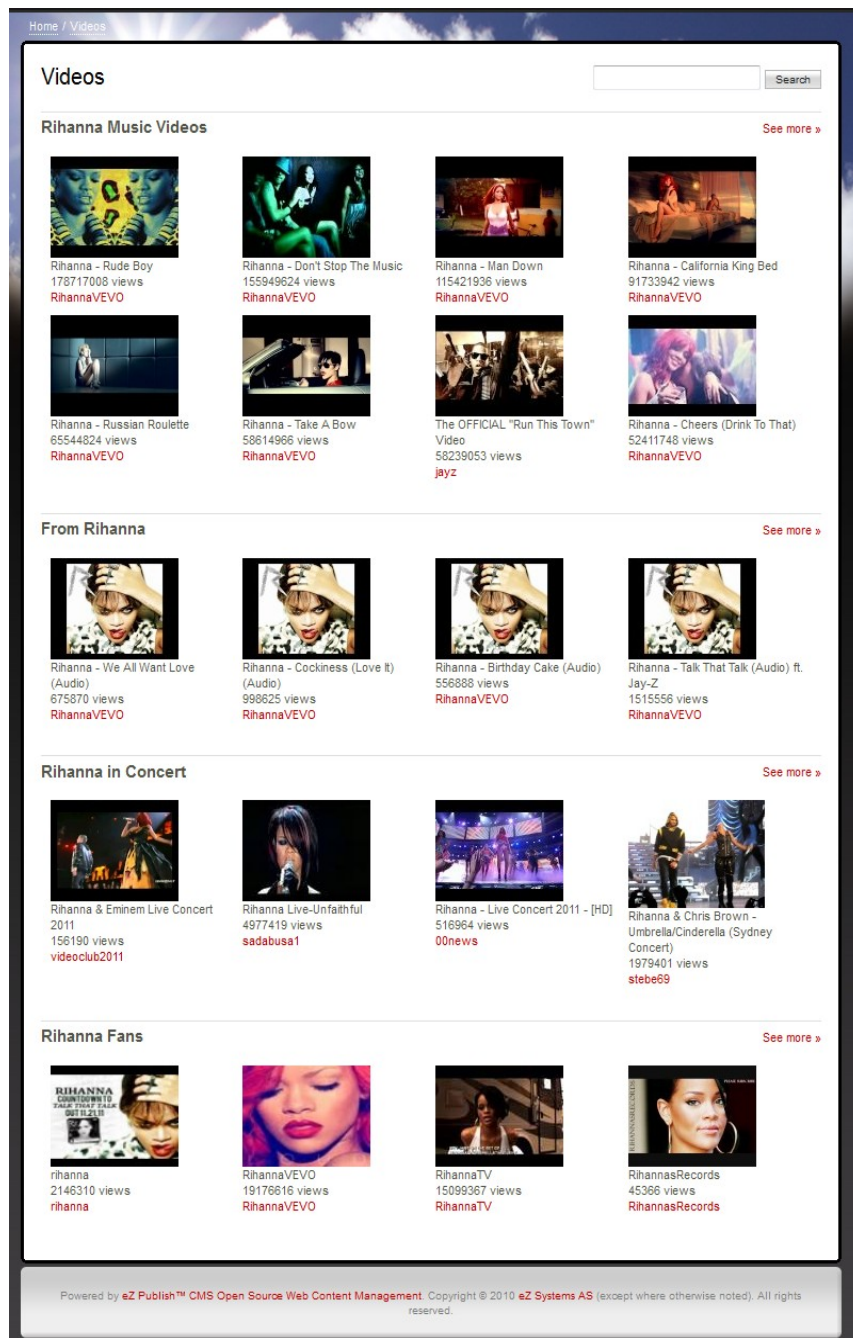
3.2 Video page

Clicking any of the video thumbnails displayed on the homepage will take you to a page that displays the selected video plus comments, video responses and related videos. In addition, information is presented about the upload date, the creator of the video, likes/dislikes, location, rating, category and tags.

3.3   Profile page

The video page also displays a link to the profile page of a YouTube user. This page displays statistics and information about the user, and presents the user uploads, playlists, favorites and contacts.
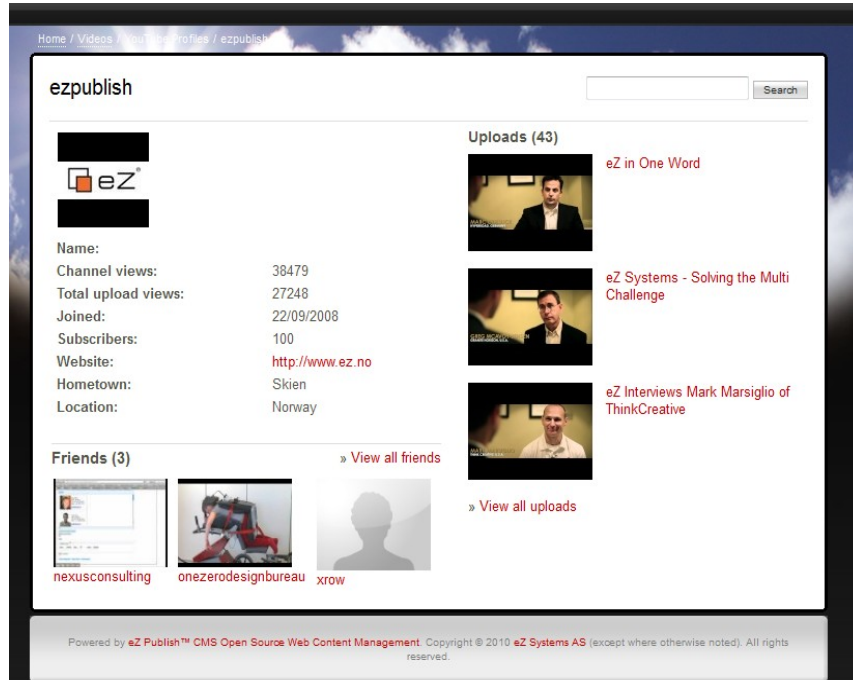
3.4   Search page

The search page (http://[sitename]/[site_access]/eztube/search) provides functionality to search for videos, profiles and playlists that match specific keywords, upload dates and/or video duration. Search results may be sorted by relevance, upload date, view count and rating:
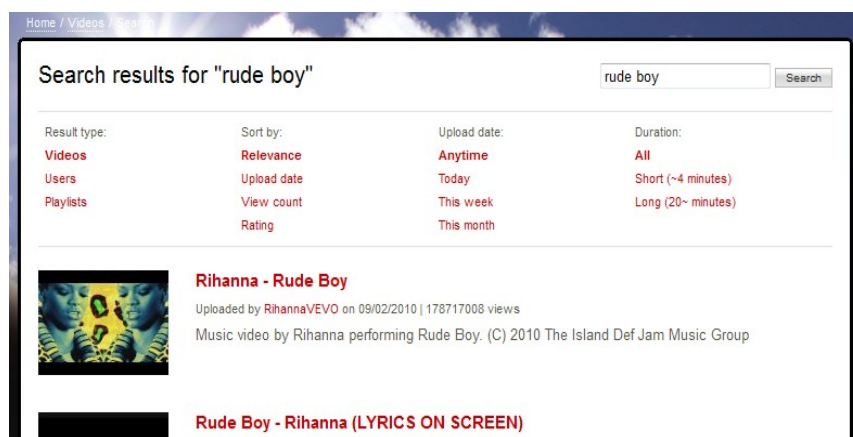
For a complete overview of all available functional pages, check section 5, "Available module views".

4.   Getting started

4.1   Content configuration

To avoid mirroring the entire YouTube.com, the eZ Tube extension comes with a number of filtering options to control the content displayed on your own website. All content filter options are defined in the configuration file 'eztube.ini'. This file allowes you control – to a certain extent – the content that is displayed on your website. The various filter options ('user', 'video', 'profile', etc.) may be set to "allow" or "deny" Setting the filter to "allow" means all items are allowed, except the ones listed under the filter. Setting the filter to "deny" means all items are denied by default, except the items listed under the filter. This allows for various degrees of restrictiveness.

For example, it is possible to deny access to all content items, except for items that have been explicitly allowed. If you have a reasonable amount of videos and you want to completely control what is displayed on your site, this is probably your best option. On the other hand, if you are the author of all videos, you can configure the filter to display only your own videos. In cases where you are not the author of the videos, and you do not want to maintain a list of allowed videos, you could "allow" all content, but filter out specific videos where necessary. It is also possible to allow only videos with a certain tag or belonging to a certain YouTube category.

> For more information on how to configure the content that is displayed on your video portal/website, check sections 7.2 and 7.3 of this document. Do not launch your site until you have proper content filters in place.

In addition to the options in the configeration file, you may further restrict access to YouTube content by limiting the access to certain eZ Tube functions through the 'policy wizard', accessible through the eZ Publish backend of your site at the URI 'role/list'.

4.2   Configuration of the homepage

By default, the homepage (eztube/home) consists of five rows. Each row displays the search results of a query defined in the configuration file *fetchalias.ini.append.php*. For example, the first two rows are defined by fetchalias '*eztube_home_row_1*':

```
[eztube_home_row_1]
Module=eztube
FunctionName=search
Constant[keywords]=rihanna|Rihanna -Eminem -clinton -Drake -Justin
Constant[filters]=true
Constant[limit]=20
Constant[order]=viewCount
Title=Rihanna Music Videos
```

The fetchalias defines a YouTube search with the search terms "rihanna" or "Rihanna", excluding search results that match the terms "Eminem", "Clinton", "Drake" and "Justin". The alias returns 20 videos, sorted by viewCount. Note that the parameter "Title" is only supported by the eZ Tube extension.

The fetchalias for the 3rd row is defined by fetchalias '*eztube_home_row_2*'. This fetchalias returns 10 videos by user/author 'RihannaVEVO". The parameter 'safeSearch' has been set to "none". The results are presented in order of publishing date.

```
[eztube_home_row_2]
Module=eztube
FunctionName=search
Constant[keywords]=
Constant[filters]=true
Constant[params]=author=RihannaVEVO;safeSearch=none
Constant[offset]=0
Constant[limit]=10
Constant[order]=published
Title=From Rihanna
```

The fetchalias for the 4th defines a YouTube search with the search terms "live concert" excluding search results that match the terms "Eminem", "Clinton", "Drake" and "Justin". The alias returns 10 videos, sorted by relevance.

```
[eztube_home_row_3]
Module=eztube
FunctionName=search
Constant[keywords]=live concert -Eminem -clinton -Drake -Justin -Chris
Constant[params]=
Constant[offset]=0
Constant[limit]=10
Constant[order]=relevance
Constant[filters]=true
Title=Rihanna in Concert
```

The fetchalias for the 5th row defines a YouTube search for users that match the search term "rihanna".

```
[eztube_home_row_4]
Module=eztube
FunctionName=search
Constant[keywords]=rihanna
Constant[searchtype]=users
Constant[params]=
Constant[offset]=0
Constant[limit]=20
Constant[filters]=true
Title=Rihanna Fans
```

The configured fetchaliases serve as an example how the homepage may be configured using the YouTube search functionality. For more information about the available configuration options for the search functionality, check section 6.11 of this document. In addition, see also Retreiving and searching for videos of "YouTube APIs and Tools". For the explanation of the use of the constant 'filters', check section 7.3 of this document. For a complete list of available paramters that can be used for the constant 'params', see "Custom parameters for the YouTube Data API", under the heading "Search parameters". Pairs of search parameter/values can be combined by adding a semicolon between the pairs, e.g:

```
Constant[params]=author=RihannaVEVO;safeSearch=none;license=cc;category=
Music
```

Of course the use of fetchaliases is not required for the functioning of the homepage. A custom homepage can also be made through a template override on home.tpl.

5. Availabe module views

The eZ Tube extension provides the following module views:

View	Function	Summary
home	home	Displays a homepage with a selection of videos
video	video	Displays a YouTube video
related	related	Displays related videos
responses	responses	Displays video responses to a video
comments	comments	Displays comments on a video
profile	profile	Displays a YouTube user profile
playlists	playlists	Displays the playlists of a user
playlist	playlist	Displays a single playlist
favorites	favorites	Displays a user's favorites
uploads	uploads	Displays a user's uploads
contacts	contacts	Displays a user's contacts
search	search	Fetches videos matching certain criteria.
tag	tag	Displays videos with similair keywords

5.1   Home page

URI
eztube/home

Description
A configurable homepage that by default consists of four rows. Each row presents the results of a search defined in fetchalias.ini.append.php.

Template
ezube/design/standard/templates/home.tpl

5.2   Video page

URI
eztube/video/[video_id]

Parameters

Name	Type	Description	Required
video_id	string	The YouTube video ID.	yes

View parameters

Name	Type	Description	Required
offset	integer	The offset to start at.	no
limit	integer	The number of videos that should be fetched.	no
order	string	The sorting mechanism that should be used.	no

Description
A page that displays a video and information about the video, as well related videos, comments and video responses.

Template
ezube/design/standard/templates/video.tpl

5.3 Related video page

URI
eztube/related/[video_id]

Parameters

Name	Type	Description	Required
video_id	string	The YouTube video ID.	yes

View parameters

Name	Type	Description	Required
offset	integer	The offset to start at.	no
limit	integer	The number of videos that should be fetched.	no
order	string	The sorting mechanism that should be used.	no

Description
A page that displays related videos.

Template
ezube/design/standard/templates/related.tpl

5.4 Video responses

URI
eztube/responses/[video_id]

Parameters

Name	Type	Description	Required
video_id	string	The YouTube video ID.	yes

View parameters

Name	Type	Description	Required
offset	integer	The offset to start at.	no
limit	integer	The number of videos that should be fetched.	no
order	string	The sorting mechanism that should be used.	no

Description
A page that displays video responses.

Template
ezube/design/standard/templates/responses.tpl

5.5 Comments

URI
eztube/comments/[video_id]

Parameters

Name	Type	Description	Required
video_id	string	The YouTube video ID.	yes

View parameters

Name	Type	Description	Required
offset	integer	The offset to start at.	no
limit	integer	The number of comments that should be fetched.	no
order	string	The sorting mechanism that should be used.	no

Description
A page that displays comments on a video.

Template
ezube/design/standard/templates/comments.tpl

5.6   Profile page

URI
eztube/profile/[profile_id]

Parameters

Name	Type	Description	Required
user_id	string	The YouTube user ID.	yes

Description
A page that displays a user profile, as well uploaded videos, contacts and playlists.

Template
ezube/design/standard/templates/profile.tpl

5.7   Playlists page

URI
eztube/playlists/[profile_id]

Parameters

Name	Type	Description	Required
user_id	string	The YouTube user ID.	yes

View parameters

Name	Type	Description	Required
offset	integer	The offset to start at.	no
limit	integer	The number of playlists that should be fetched.	no
order	string	The sorting mechanism that should be used.	no

Description
A page that displays the playlists of a user.

Template
ezube/design/standard/templates/playlists.tpl

5.8   Playlist page

URI
eztube/playlist/[playlist_id]

Parameters

Name	Type	Description	Required
playlist_id	string	The YouTube playlist ID.	yes

View parameters

Name	Type	Description	Required
offset	integer	The offset to start at.	no
limit	integer	The number of videos that should be fetched.	no
order	string	The sorting mechanism that should be used.	no

Description
A page that displays the videos belonging to a specific playlist.

Template
ezube/design/standard/templates/playlist.tpl

5.9   Favorites page

URI
eztube/favorites/[user_id]

Parameters

Name	Type	Description	Required
user_id	string	The YouTube user ID.	yes

View parameters

Name	Type	Description	Required
offset	integer	The offset to start at.	no
limit	integer	The number of favorites that should be fetched.	no
order	string	The sorting mechanism that should be used.	no

Description
A page that displays the favorites of a user.

Template
ezube/design/standard/templates/favorites.tpl

5.10 Uploads page

URI
eztube/uploads/[user_id]

Parameters

Name	Type	Description	Required
user_id	string	The YouTube user ID.	yes

View parameters

Name	Type	Description	Required
offset	integer	The offset to start at.	no
limit	integer	The number of uploads that should be fetched.	no
order	string	The sorting mechanism that should be used.	no

Description

A page that displays the uploaded videos by a user.

Template

ezube/design/standard/templates/uploads.tpl

5.11 Contacts page

URI

eztube/contacts/[user_id]

Parameters

Name	Type	Description	Required
user_id	string	The YouTube user ID.	yes

View parameters

Name	Type	Description	Required
offset	integer	The offset to start at.	no
limit	integer	The number of contacts that should be fetched.	no
order	string	The sorting mechanism that should be used.	no

Description

A page that displays the contacts of a user.

Template

ezube/design/standard/templates/contacts.tpl

5.12 Search page

URI
eztube/search

View parameters

Name	Type	Description	Required
offset	integer	The offset to start at.	no
limit	integer	The number of videos that should be fetched.	no
order	string	The sorting mechanism that should be used.	no

Get variables

Name	Type	Description	Required
q	string	The text that should be matched.	no
search_type	string	Search 'videos', 'users' or 'playlists'.	no
time	string	Upload date of the video	no
duration	string	Duration of the video	no
author	string	Author of the video	no

Additional Get variables may be defined under the 'AllowedSearchParameters' settings group in eztube.ini.

Description
A page that displays search results.

Template
ezube/design/standard/templates/search.tpl

5.13 Tag page

URI
eztube/tag/[tag]

Parameters

Name	Type	Description	Required
tag	string	YouTube tag	yes

View parameters

Name	Type	Description	Required
offset	integer	The offset to start at.	no
limit	integer	The number of videos that should be fetched.	no
order	string	The sorting mechanism that should be used.	no

Description
A page that displays videos that are tagged with the selected term.

Template
ezube/design/standard/templates/tag.tpl

6.   Availabe template functions

The eZ Tube extension offers a collection of various [functions](#) that can be used to carry out different tasks. The following functions are available:

Function	Summary
video	Fetches a YouTube video
related	Fetches related videos
responses	Fetches video responses to a video
comments	Fetches comments on a video
profile	Fetches a YouTube user profile
playlists	Fetches the playlists of a user
playlist	Fetches a single playlist
favorites	Fetches a user's favorites
uploads	Fetches a user's uploads
contacts	Fetches a user's contacts
search	Fetches videos matching certain criteria.
standard	Fetches standard YouTube feeds
categories	Fetches a list of YouTube categories

6.1   Video

Summary
Fetches a YouTube video.

Usage

```
fetch( 'eztube', 'video', hash( [ 'video_id', video_id, ]
                                 [ 'minimal', minimal ]     ))
```

Parameters

Name	Type	Description	Required
video_id	string	The YouTube video ID.	yes
minimal	boolean	Fetch all available info or only a subset.	no

Returns
An array with video information or FALSE.

Description
This function fetches a YouTube video. If mimimal is set to false, it also returns array with 25 comments, related videos and video responses.

YouTube API documentation

Display information about a video

Example

```
{def $video = fetch( 'eztube', 'video', hash(
                    'video_id', 'H-0OuRVCAK8',
                    'minimal', false ) )}

<strong>{$video.title}</strong><br />

{foreach $video.comments.comments_list as $comment}
     {$comment.title|wash}<br />
{/foreach}
```

6.2   Related

Summary
Fetches a list of related videos.

Usage

```
fetch( 'eztube', 'related', hash( [ 'video_id', video_id, ]
                                  [ 'offset', offset, ]
                                  [ 'limit', limit, ]
                                  [ 'order', order ]          ))
```

Parameters

Name	Type	Description	Required
video_id	string	The video ID.	yes
offset	integer	The offset to start at.	no
limit	integer	The number of videos that should be fetched.	no
order	string	The sorting mechanism that should be used.	no

Returns
An array with related videos or FALSE.

Description
This function fetches a list of related videos.

YouTube API documentation

Related videos

Example

```
{def $related_videos = fetch( 'eztube', 'related', hash(
                              'video_id', 'H-0OuRVCAK8',
                              'offset', 0, 'limit', 10 ) )}

{foreach $related_videos.video_list as $video}
     {$video.title|wash}<br />
{/foreach}
```

6.3   Responses

Summary
Fetches a list of video responses.

Usage

```
fetch( 'eztube', 'responses', hash( [ 'video_id', video_id, ]
                                    [ 'offset', offset, ]
                                    [ 'limit', limit, ]
                                    [ 'order', order ]          ))
```

Parameters

Name	Type	Description	Required
video_id	string	The video ID.	yes
offset	integer	The offset to start at.	no
limit	integer	The number of videos that should be fetched.	no
order	string	The sorting mechanism that should be used.	no

Returns
An array with video responses or FALSE.

Description
This function fetches a list of video responses.

YouTube API documentation

Video responses

Example

```
{def $video_responses = fetch( 'eztube', 'responses', hash(
                               'video_id', 'WdgLMslbDuY',
                               'offset', 0, 'limit', 10 ) )}

{foreach $video_responses.video_list as $video}
     {$video.title|wash}<br />
{/foreach}
```

6.4   Comments

Summary
Fetches a list of comments on a video.

Usage

```
fetch( 'eztube', 'comments', hash( [ 'video_id', video_id, ]
                                   [ 'offset', offset, ]
                                   [ 'limit', limit, ]
                                   [ 'order', order ]              ))
```

Parameters

Name	Type	Description	Required
video_id	string	The video ID.	yes
offset	integer	The offset to start at.	no
limit	integer	The number of videos that should be fetched.	no
order	string	The sorting mechanism that should be used.	no

Returns
An array with comments or FALSE.

Description
This function fetches a list of comments on a video.

YouTube API documentation

Comments

Example

```
{def $comments = fetch( 'eztube', 'comments', hash(
                        'video_id', '8iFYJYwtjpc',
                        'offset', 0, 'limit', 10 ) )}

{$comments.comments_count} comments:<br/>

{foreach $comments.comments_list as $video}
     {$video.title|wash} <br />
{/foreach}
```

6.5   Profile

Summary
Fetches a YouTube user profile.

Usage

```
fetch( 'eztube', 'profile', hash( [ 'user_id', user_id, ]
                                   [ 'minimal', minimal ]      ))
```

Parameters

Name	Type	Description	Required
user_id	string	The YouTube user ID.	yes
minimal	boolean	Fetch all available info or only a subset.	no

Returns
An array with profile information or FALSE.

Description

This function fetches a YouTube profile. If mimimal is set to false, it also returns array with 25 video uploads and user contacts.

YouTube API documentation

User profiles

Example

```
{def $profile = fetch( 'eztube', 'profile', hash(
                       'user_id', 'RihannaVEVO',
                       'minimal', false ) )}

<strong>{$profile.user_name|wash}</strong><br />
```

6.6   Playlists

Summary

Fetches the playlists created by a YouTube user.

Usage

```
fetch( 'eztube', 'playlists', hash( [ 'user_id', user_id, ]
                                     [ 'offset', offset, ]
                                     [ 'limit', limit, ]      ))
```

Parameters

Name	Type	Description	Required
user_id	string	The YouTube user ID.	yes
offset	integer	The offset to start at.	no
limit	integer	The number of videos that should be fetched.	no

Returns

An array with user information and playlists or FALSE.

Description

This function fetches the playlists by a YouTube user. It also fetches the user information of the

user that has created the playlists.

YouTube API documentation

[Retrieving a user's playlists](#)

Example

```
{def $playlists = fetch( 'eztube', 'playlists', hash(
                         'user_id', 'RihannaVEVO',
                         'offset', 0, 'limit', 10 ) )}

{foreach $playlists_list as $playlist}
      {$playlist.title|wash}<br />
{/foreach}
```

6.7   Playlist

Summary
Fetches a single playlist created by a YouTube user.

Usage

```
fetch( 'eztube', 'playlist', hash( [ 'playlist_id', playlist_id, ]
                                   [ 'offset', offset, ]
                                   [ 'limit', limit, ]
                                   [ 'order', order ]              ))
```

Parameters

Name	Type	Description	Required
playlist_id	string	The playlist ID.	yes
offset	integer	The offset to start at.	no
limit	integer	The number of videos that should be fetched.	no
order	string	The sorting mechanism that should be used.	no

Returns
FALSE or an array with user and playlist information and a list of videos in the playlist.

Description
This function fetches a single playlist by a YouTube user. It also fetches the user information of the user that created the playlist.

YouTube API documentation

[Retrieving a single playlist](#)

Example

```
{def $user_playlist = fetch( 'eztube', 'playlist', hash(
                             'playlist_id', '898704AC68C23F1D',
                             'offset',0, 'limit', 2,
                             'order', 'duration' ) )}

{foreach $user_playlist.video_list as $video}
      {$video.title|wash}<br />
{/foreach}
```

6.8 Favorites

Summary
Fetches the favorites created by a YouTube user.

Usage

```
fetch( 'eztube', 'favorites', hash( [ 'user_id', user_id, ]
                                    [ 'offset', offset, ]
                                    [ 'limit', limit, ],
                                    [ 'order', order ]      ))
```

Parameters

Name	Type	Description	Required
user_id	string	The user ID.	yes
offset	integer	The offset to start at.	no
limit	integer	The number of videos that should be fetched.	no
order	string	The sorting mechanism that should be used.	no

Returns
FALSE or an array with user and playlist information and a list of videos in the playlist.

Description

This function returns the favorites of a YouTube user. It also fetches the user information of the user that the favorites belong to.

YouTube API documentation

Retrieving favorites

Example

```
{def $favorites = fetch( 'eztube', 'favorites', hash(
                         'user_id', 'RihannaVEVO' ,
                         'offset', 0, 'limit', 10 ) )}

{foreach $favorites.video_list as $favorite}
      {$favorite.title|wash} <br />
{/foreach}
```

6.9   Uploads

Summary

Fetches the videos uploaded by a YouTube user.

Usage

```
fetch( 'eztube', 'uploads', hash( [ 'user_id', user_id, ]
                                  [ 'offset', offset, ]
                                  [ 'limit', limit, ],
                                  [ 'order', order ]       ))
```

Parameters

Name	Type	Description	Required
user_id	string	The user ID.	yes
offset	integer	The offset to start at.	no
limit	integer	The number of videos that should be fetched.	no
order	string	The sorting mechanism that should be used.	no

Returns

FALSE or an array with user and upload information and a list of uploaded videos.

Description
This function returns the uploads of a YouTube user. It also fetches the user information of the user that the uploads belong to.

YouTube API documentation

Videos uploaded by a specific user

Example

```
{def $uploads = fetch( 'eztube', 'uploads', hash(
                       'user_id', 'RihannaVEVO' ,
                       'offset', 0, 'limit', 25 ) )}

{foreach $uploads.video_list as $upload}
     {$upload.title|wash} <br />
{/foreach}
```

6.10 Contacts

Summary
Fetches a user's contacts.

Usage

```
fetch( 'eztube', 'contacts', hash( [ 'user_id', user_id, ]
                                    [ 'offset', offset, ]
                                    [ 'limit', limit, ],
                                    [ 'order', order ]      ))
```

Parameters

Name	Type	Description	Required
user_id	string	The user ID.	yes
offset	integer	The offset to start at.	no
limit	integer	The number of videos that should be fetched.	no
order	string	The sorting mechanism that should be used.	no

Returns

FALSE or an array with contacts and profile information of the user.

Description

This function returns the contacts ('profiles') of a YouTube user and profile information about the YouTube user.

YouTube API documentation

Contacts

Example

```
{def $contacts = fetch( 'eztube', 'contacts', hash(
                        'user_id', 'ezpublish' ,
                        'offset', 0, 'limit', 25 ) )}

{foreach $contacts.contacts_list as $contact}
     {$contact.display_name|wash} <br />
{/foreach}
```

6.11 Search

Summary

Fetches videos matching certain criteria.

Usage

```
fetch( 'eztube', 'search', hash( [ 'keywords', keywords, ]
                                 [ 'searchtype', searchtype, ]
                                 [ 'filters', filters, ],
                                 [ 'params', params ],
                                 [ 'offset', offset ],
                                 [ 'limit', limit ],
                                 [ 'order', order ]              ))
```

Parameters

Name	Type	Description	Required
keywords	string	The text that should be matched.	no
searchtype	string	Search 'videos', 'users' or 'playlists'.	no
filters	boolean	Use filters specified supplied in the 'StandardSearchFilters' group in eztube.ini	no
params	array	Additional search parameters	no
offset	integer	The offset to start at.	no
limit	integer	The number of videos that should be fetched.	no
order	string	The sorting mechanism that should be used.	no

Returns

FALSE or an array with videos.

Description

This function returns an array with videos, channels or playlists that match the search criteria. For the explanation of the use of the parameter 'filters', please check section 7.3 of this document.

YouTube API documentation

[Retreiving and searching for videos](#)

Example 1

Find videos uploaded by user 'RihannaVEVO', including restricted content.

```
{def $params = hash('author', 'RihannaVEVO','safeSearch','none')}

{def $searchresults = fetch( 'eztube', 'search', hash(
                             'params', $params, 'offset', 0,
                             'limit', 25, 'order', 'published' ) )}

{foreach $searchresults.result_list as $item}
     {$item.title|wash} <br />
{/foreach}
```

Example 2

Find videos with meta data that contains the search terms 'rihanna', 'concert' and 'Sydney', and order the results by viewCount.

```
{def $searchresults = fetch( 'eztube', 'search', hash(
                             'keywords', '"rihanna concert Sydney"',
                             'offset', 0, 'limit', 25,
                             'order', 'viewCount' ) )}
```

Example 3

Find users that match the search terms 'rihanna fan' and order the results by relevance.

```
{def $searchresults = fetch( 'eztube', 'search', hash(
                            'keywords', 'rihanna fan',
                            'searchtype', 'users',
                            'offset', 0, 'limit', 5,
                            'order', 'relevance' ) )}
```

Example 4

Find playlists that match the search terms 'rihanna' and order the results by publish date.

```
{def $searchresults = fetch( 'eztube', 'search', hash(
                            'keywords', 'rihanna',
                            'searchtype', 'playlists',
                            'offset', 0, 'limit', 5,
                            'order', 'published' ) )}
```

6.12 Standard feeds

Summary

Fetches standard YouTube feeds.

Usage

```
fetch( 'eztube', 'standard', hash( [ 'type', type, ]
                                   [ 'time', time, ]
                                   [ 'region', region, ],
                                   [ 'category', category ],
                                   [ 'limit', limit ]        ))
```

Parameters

Name	Type	Description	Required
type	string	Type of standard feed (see: 'feed types')	yes
time	string	Search 'day', 'week', 'month'.	no
region	string	Region specific feed (for example: 'US', 'GB')	no
category	array	Category specific feed (for example: 'Music')	no
limit	integer	The number of videos that should be fetched.	no

Feed types

Name	Description
rated	Top rated
favorites	Top favorites
viewed	Most viewed
shared	Most shared
recent	Most recent
responded	Most responded
recently	Recently featured
discussed	Most discussed
trending	Trending videos
popular	Most popular (default)

Returns
FALSE or an array with videos.

Description
This function returns an array with videos.

YouTube API documentation

[Standard video feeds](#)

Example 1
List of today's most shared Tech videos in Australia.

```
{def $popular = fetch( 'eztube', 'standard', hash(
                    'type', 'shared', 'time','today',
                    'region', 'AU', 'category', 'Tech',
                    'limit', 25 ) )}

{foreach $popular.video_list as $item}
     {$item.title|wash} <br />
{/foreach}
```

6.12 Categories

Summary

Fetches a list with (localized) YouTube categories.

Usage

```
fetch( 'eztube', 'categories', hash( [ 'language', language ] ))
```

Parameters

Name	Type	Description	Required
language	string	language	no

Returns
FALSE or an array with categories.

Description
This function returns an array with categories.

YouTube API documentation

Localized category lists

Example 1
List of Dutch categories.

```
{def $categories = fetch( 'eztube', 'categories', hash(
                    'language', 'nl-NL' ) )}

{$categories.category_list|attribute(show)}
```

Example 2
List of Dutch categories.

```
{def $categories = fetch( 'eztube', 'categories')}

{$categories.category_list|attribute(show)}
```

6.13 Sorting

The "order" parameter makes it possible to sort the result in different ways. The following table shows the sorting methods that can be used for most lists.

Sorting method	Description
relevance	Entries are ordered by their relevance to a search query.
published	Entries are returned in reverse chronological order.
viewCount	Entries are ordered from most views to least views.
rating	Entries are ordered from highest rating to lowest rating.

In a request for a playlist feed, the following values are valid for the "order" parameter:

Sorting method	Description
position	Entries are ordered by their position in the playlist. This is the default setting.
commentCount	Entries are ordered by number of comments from most comments to least comments.
duration	Entries are ordered by length of each playlist video from longest video to shortest video.
published	Entries are returned in reverse chronological order.
title	Entries are ordered alphabetically by title.
viewCount	Entries are ordered from most views to least views.

7. Settings

7.1 Cache settings

XML files and images retrieved through the YouTube API are cached in the folder 'eztube' in the cache directory. Cache settings are available in the settings file '*eztube.ini*'.

XML files are cached in the subdirectory 'XML'. Separate subdirectories are created for cached XML files containing information about videos, users and comments. Other types of XML files, such as playlists, standard feeds, etc. are cached in the subdirectory 'other'. The TTL may be specified per type of cache. By default, xml files for videos and users are refresed 24 hours after creation. The cache for comments expires after 60 minutes, in order to ensure new comments display without too much delay. Caching of XML files can not be disabled. It is possible to create seperate cache files for playlists and standard feeds. Please check eztube.ini for more info.

The eztube cache may be cleared through the administrative backend of eZ Publish ('Cache management'). Under the heading 'Fine-grained Cache Control', the following eZ Tube clear cache options are available:

- eZ Tube images;
- eZ Tube XML cache – comments;
- eZ Tube XML cache – videos;
- eZ Tube XML cache – users; and
- eZTube XML cache – other.

Alternatively, the cache files may be expired by running the following command from the command prompt with one of the following identifiers: *eztube-xml-comments*, *eztube-xml-videos*, *eztube-xml-users*, *eztube-images* or *eztube-xml-other*:

```
php bin/php/ezcache.php --clear-id=[identifier]
```

Please note: eZ Tube uses the "stale cache" approach. This means that "instead of deleting cache elements and then regenerate them across concurrent requests, the logic is reversed to a refresh algorithm. Basically a cache element will be marked as invalid but not deleted until a new version is ready. During the generation of the new cache element, the old version will still be served until it is ready." (Nicolas Pastorino, "eZ Publish Knowledge Series: «Stale Cache» or «How caches in eZ Publish 4.1 are handled in a smarter way»"). Clearing the cache will not delete the cached xml and image files from your disk. Instead you need to run the following script with one of the identifiers mentioned above to purge your cache files and stay in control of the disk space used. You should set up these scripts as cronjobs.

```
php bin/php/ezcache.php --clear-id=[identifier]  --purge
```

Images are cached in the subdirectory 'images'. Unlike XML caching, image caching may be disabled. The YouTube API provides a number of image variations: *default.jpg*, *hqdefault.jpg*, *0.jpg*, *1.jpg*, *2.jpg* and *3.jpg*. By default, only *default.jpg* is cached and used in the extension templates. The file *hqdefault.jpg* provides a high quality version of the default images. The other files are various stills from the video. It is possible to also enable the caching of each of these image variations. However: enabling caching of all available image variations will increase the load on your server, and may eventually cause disk space problems.

Setting the TTL to low may cause additional load on the server because it has to retrieve files from YouTube servers. Setting the TTL too high may eventually cause disk space issues, especially in the case of images. It is strongly recommended you monitor the disk space used for caching on a frequent and regular basis. Make sure you have cronjobs set up to purge expired cache frequently.

For more information about how caching is implemented in the eZ Tube extension, check the following blog post: "How to implement caching for extensions".

7.2   Content Filter settings

The Filter rules in the configuration file 'eztube.ini' allow you to control – to a certain extent – the content that is displayed on your website. If you do not specify filters at all, any video or user profile on YouTube may be displayed on your site, including videos with content that you might not approve of. The following filter options are available:

- Video filter;

- User filter;

- Profile filter;

- Category filter; and

- Tag filter.

Each of these filters may be set to "allow" or "deny". Setting the filter to "allow" means all items are allowed, except the ones listed under the filter. Setting the filter to "deny" means all items are denied by default, except the items listed under the filter. For example, the setting below allows the display of all videos, except the video with ID "rJYcmq__nDM" (Rihanna - Rehab ft. Justin Timberlake).

```
VideoFilterRule=allow
VideoFilter[]=rJYcmq__nDM
```

The category and tag filters filter videos based on the YouTube categories selected, or the tags assigned to a video. For example, the setting below allows all videos, except the ones that have been labelled with the specified tags:

```
TagFilterRule=allow
TagFilter[]=timberlake
TagFilter[]=Cover
TagFilter[]=Acoustic
```

Users may be filtered by using the 'UserFilterRule' setting or the 'ProfileFilterRule' setting. The User Filter filters user profiles on basis of the YouTube user ID. For example, the setting below denies all users, except 'RihannaVEVO'.

```
UserFilterRule=deny
UserFilter[]=RihannaVEVO
```

The Profile filter checks if a specific user has published any items that match the search terms listed. For example, the setting below denies the display of all profiles, except the ones where the users have published an item that matches the search term 'Rihanna'.

```
ProfileFilterRule=deny
ProfileFilter[]=rihanna
```

Due to API limitations, these filters are applied on the search results, and not as part of the search. Therefore the actual search result count may differ from the result count provided by YouTube. In some cases this may cause missing records in the search results: instead of displaying 25 search records per page, some pages may display less than 25 records. By enabling the 'UseAutoFill' setting under the settings group 'SearchSettings', missing records are replaced by records from the next set of search results. However, this may cause duplication of search results across the following pages.

> Even with the various filters in place it is still possible that content from YouTube will be displayed on your website that you, or your users, do not approve of. It is therefore strongly recommended you monitor the YouTube content displayed on your site, and apply additional filters where needed. Please note: If you do not specify filters at all, any video or user profile on YouTube may be displayed as an integral part of your site.

7.3   Standard Search Filters

In addition to the filters described above, it is also possible to define search parameters that are automatically appended to all searches, and can not be overruled by the users of the site. These search filters may be specified under the 'StandardSearchFilters' settings group. Available parameters include geolocation, duration, license, safeSearch, etc. For a complete lists of available search parameters, check "Custom parameters for the YouTube Data API", under the heading "Search parameters".

The 'q' parameter specifies the search query term(s). YouTube will search all video metadata for videos matching the term. Video metadata includes titles, keywords, descriptions, authors' usernames, and categories. To search for an exact phrase, enclose the phrase in quotation marks. Your request can also use the Boolean NOT (-) and OR (|) operators to exclude videos or to find

videos that are associated with one of several search terms. For example, the phrase below searches for "eZ Publish" or "ezpublish", but excludes search results that also contain "Wordpress":

```
[StandardSearchFilters]
Filter[q]="ez publish"|ezpublish+-Wordpress
```

By specifiying this filter, the phrase below is automatically added to all user defined searches. Therefore, if a user searches for "open source", the search query will automatically include "ez publish" and "ezpublish" as additional keywords, while excluding all search results containing the term "Wordpress". Adding filters to the 'StandardSearchFilters' settings group does not automatically make these settings available for the users as well. Add the parameters to the 'AllowedSearchParameters' settings group if you want to allow users to use these parameters. Please note that ' AllowedSearchParameters' do not show up automatically in the search template. You will need to modify the search.tpl template if you want to present these search options to the users.

Content search filters are applied in addition to the standard search filters.

It is strongly recommended that you do not change the default setting 'strict' of the 'safeSearch' parameter.

7.4   YouTube Embedded Player Parameters

The 'PlayerSettings' settings group defines the parameters that are available in the YouTube embedded player. By modifying or adding these parameters, you can customize the playback experience in your application. For example, you can automatically play videos using the autoplay parameter or cause a video to play repeatedly using the loop parameter. More information about the available parameters is available at: "YouTube embedded player parameters".