
youtube-dl- www.ssyoutube.com

Autor:

Data de publicació: 02-08-2014

<https://www.ssyoutube.com/watch?v=SXfnm3Bm13Q>

Documentation

Options:

General Options:

-h, --help print this help text and exit
--version print program version and exit
-U, --update update this program to latest version. Make sure
that you have sufficient permissions (run with
sudo if needed)
-i, --ignore-errors continue on download errors
--dump-user-agent display the current browser identification
--user-agent UA specify a custom user agent
--referer REF specify a custom referer, use if the video access
is restricted to one domain
--list-extractors List all supported extractors and the URLs they
would handle
--extractor-descriptions Output descriptions of all supported extractors
--proxy URL Use the specified HTTP/HTTPS proxy
--no-check-certificate Suppress HTTPS certificate validation.

Video Selection:

--playlist-start NUMBER playlist video to start at (default is 1)
--playlist-end NUMBER playlist video to end at (default is last)
--match-title REGEX download only matching titles (regex or caseless
sub-string)
--reject-title REGEX skip download for matching titles (regex or
caseless sub-string)
--max-downloads NUMBER Abort after downloading NUMBER files
--min-filesize SIZE Do not download any videos smaller than SIZE
(e.g. 50k or 44.6m)
--max-filesize SIZE Do not download any videos larger than SIZE (e.g.
50k or 44.6m)
--date DATE download only videos uploaded in this date
--datebefore DATE download only videos uploaded before this date
--dateafter DATE download only videos uploaded after this date

Download Options:

-r, --rate-limit LIMIT maximum download rate (e.g. 50k or 44.6m)
-R, --retries RETRIES number of retries (default is 10)
--buffer-size SIZE size of download buffer (e.g. 1024 or 16k)
(default is 1024)
--no-resize-buffer do not automatically adjust the buffer size. By

default, the buffer size is automatically resized from an initial value of SIZE.

Filesystem Options:

-t, --title use title in file name (default)
--id use only video ID in file name
-l, --literal [deprecated] alias of --title
-A, --auto-number number downloaded files starting from 00000
-o, --output TEMPLATE output filename template. Use %(title)s to get the title, %(uploader)s for the uploader name, %(uploader_id)s for the uploader nickname if different, %(autonumber)s to get an automatically incremented number, %(ext)s for the filename extension, %(upload_date)s for the upload date (YYYYMMDD), %(extractor)s for the provider (youtube, metacafe, etc), %(id)s for the video id, %(playlist)s for the playlist the video is in, %(playlist_index)s for the position in the playlist and %% for a literal percent. Use - to output to stdout. Can also be used to download to a different directory, for example with -o '/my/downloads/%(uploader)s/%(title)s-%(id)s.%(ext)s' .
--autonumber-size NUMBER Specifies the number of digits in %(autonumber)s when it is present in output filename template or --autonumber option is given
--restrict-filenames Restrict filenames to only ASCII characters, and avoid "&" and spaces in filenames
-a, --batch-file FILE file containing URLs to download ('-' for stdin)
-w, --no-overwrites do not overwrite files
-c, --continue resume partially downloaded files
--no-continue do not resume partially downloaded files (restart from beginning)
--cookies FILE file to read cookies from and dump cookie jar in
--no-part do not use .part files
--no-mtime do not use the Last-modified header to set the file modification time
--write-description write video description to a .description file
--write-info-json write video metadata to a .info.json file
--write-thumbnail write thumbnail image to disk

Verbosity / Simulation Options:

-q, --quiet activates quiet mode
-s, --simulate do not download the video and do not write anything to disk
--skip-download do not download the video
-g, --get-url simulate, quiet but print URL
-e, --get-title simulate, quiet but print title
--get-id simulate, quiet but print id
--get-thumbnail simulate, quiet but print thumbnail URL
--get-description simulate, quiet but print video description
--get-filename simulate, quiet but print output filename
--get-format simulate, quiet but print output format
--newline output progress bar as new lines
--no-progress do not print progress bar
--console-title display progress in console titlebar
-v, --verbose print various debugging information
--dump-intermediate-pages print downloaded pages to debug problems(verbose)

Video Format Options:

-f, --format FORMAT video format code, specify the order of preference using slashes: "-f 22/17/18"

--all-formats download all available video formats

--prefer-free-formats prefer free video formats unless a specific one is requested

--max-quality FORMAT highest quality format to download

-F, --list-formats list all available formats (currently youtube only)

Subtitle Options:

--write-sub write subtitle file (currently youtube only)

--write-auto-sub write automatic subtitle file (currently youtube only)

--only-sub [deprecated] alias of --skip-download

--all-sub downloads all the available subtitles of the video

--list-sub lists all available subtitles for the video

--sub-format FORMAT subtitle format (default=srt) ([sbv/vtt] youtube only)

--sub-lang LANGS languages of the subtitles to download (optional) separated by commas, use IETF language tags like 'en,pt'

Authentication Options:

-u, --username USERNAME account username

-p, --password PASSWORD account password

-n, --netrc use .netrc authentication data

--video-password PASSWORD video password (vimeo only)

Post-processing Options:

-x, --extract-audio convert video files to audio-only files (requires ffmpeg or avconv and fprobe or avprobe)

--audio-format FORMAT "best", "aac", "vorbis", "mp3", "m4a", "opus", or "wav"; best by default

--audio-quality QUALITY ffmpeg/avconv audio quality specification, insert a value between 0 (better) and 9 (worse) for VBR or a specific bitrate like 128K (default 5)

--recode-video FORMAT Encode the video to another format if necessary (currently supported: mp4|flv|ogg|webm)

-k, --keep-video keeps the video file on disk after the post-processing; the video is erased by default

--no-post-overwrites do not overwrite post-processed files; the post-processed files are overwritten by default

--embed-sub embed subtitles in the video (only for mp4 videos)

Index

Getting started
More usage tips
Frequently Asked Questions
Supported sites
Proxy support
Video formats
Output template
Configuration

Getting started

In Windows save the exe version somewhere (in your PATH if you want to run it simply as youtube-dl) and run it from the command line.

In Unix, download it, give it execution permission and copy it to one of the PATH directories (typically, /usr/local/bin).

To update, simply call youtube-dl -U, we recommend to do it often.

After that, you should be able to call it from the command line as youtube-dl. I will use youtube-dl in the following examples. Usage instructions are easy. Use youtube-dl followed by a video URL or identifier. Example: youtube-dl "http://www.youtube.com/watch?v=foobar". The video will be saved to the file foobar.flv in that example. As YouTube.com videos are usually in Flash Video format, their extension should be flv. In Linux and other unices, video players using a recent version of ffmpeg can play them. That includes VLC, MPlayer, etc. Those two work under Windows and other platforms, but you could also get a specific FLV player of your taste.

If you try to run the program and you receive an error message containing the keyword SyntaxError near the end, it means your Python interpreter is too old, remember that youtube-dl need Python version 2.6, 2.7, or 3.3+.

More usage tips

The program is usually invoked as youtube-dl followed by options and the video URLs. Listing all the options here would make this text too long, so you can run youtube-dl --help and get a summary of them. From that point on you can start experimenting with the different options yourself. The most common ones are -t (or -l) to include the video title in the file name. Also, the -o option can specify the output file name and path. It allows special character sequences that can be used as templates to be replaced. See the "Output template" section for more details.

Some YouTube videos are served using Adobe's proprietary RTMP protocol, which imposes DRM restrictions and encrypts the connection. youtube-dl is not able to download these videos by itself. However, it will attempt to use the rtmpdump program, if it's installed in your system, to download these videos.

Frequently Asked Questions

The Frequently Asked Questions page contains answers to some common questions that pop up in my e-mail and in the issue tracker. Be sure to check it before reporting problems.

Supported sites

youtube-dl can download videos from not only YouTube, but over 150 different services. For a complete list, see the Supported sites pages or execute youtube-dl --extractor-descriptions.

Proxy support

youtube-dl supports downloading videos through a proxy, by setting the http_proxy environment variable to the proxy URL, as in http://proxy_machine_name:port/.

YouTube Video

Using the -f option and other related options, you can specify the video format to be downloaded. Add the -F option to see all valid formats. If you have an order of preference other than best quality, specify the formats separating them with slashes: -f 22/17/18. You can also pass in an extension, like -f mp4/webm.

Output template

The -o option allows users to indicate a template for the output file names. The basic usage is not to set any template arguments when downloading a single file, like in youtube-dl -o funny_video.flv "http://some/video". However, it may contain special sequences that will be replaced when downloading each video. The special sequences have the format %(NAME)s. To clarify, that's a percent symbol followed by a name in parenthesis, followed by a lowercase S. Allowed names are:

id: The sequence will be replaced by the video identifier.

url: The sequence will be replaced by the video URL.

uploader: The sequence will be replaced by the nickname of the person who uploaded the video.

upload_date: The sequence will be replaced by the upload date in YYYYMMDD format.

title: The sequence will be replaced by the video title.

ext: The sequence will be replaced by the appropriate extension (like flv or mp4).

epoch: The sequence will be replaced by the Unix epoch when creating the file.

autonumber: The sequence will be replaced by a five-digit number that will be increased with each download, starting at zero.

The current default template is `%(id)s.%(ext)s`, but that will be switched to `%(title)s-%(id)s.%(ext)s` (which can be requested with `-t` at the moment).

In some cases, you don't want special characters such as `?`, spaces, or `&`, such as when transferring the downloaded filename to a Windows system or the filename through an 8bit-unsafe channel. In these cases, add the `--restrict-filenames` flag to get a shorter title:

```
$ youtube-dl --get-filename -o "%(title)s.%(ext)s" BaW_jenozKc youtube-dl test video "_ä?
```