
youtube-dl-api-server

Release 0.3

Aug 03, 2017

Contents

1	Install the server	3
1.1	Using pip	3
1.2	From source	3
1.3	App Engine	3
1.4	Heroku	3
2	youtube-dl-server	5
3	API	7
3.1	API methods	7
3.2	Test server	11
4	youtube-dl-api-server Changelog	13
4.1	Version 0.3	13
4.2	Version 0.2	13
4.3	Version 0.1.2	13
4.4	Version 0.1.1	14
4.5	Version 0.1	14
5	About	15
5.1	Authors	15
5.2	License	15
6	Indices and tables	17
	HTTP Routing Table	19

Contents:

CHAPTER 1

Install the server

How to install **youtube-dl-api-server**

Using pip

Just run **pip install --pre youtube_dl_server**.

Then you can start using the server with **youtube-dl-server**.

From source

Download the source code and install the dependencies with **pip install -r requirements.txt**. Then just run **python -m youtube_dl_server**

Alternatively you can use pip to install it, run **pip install -e .** and then you'll be able to run **youtube-dl-server** (you won't need to rerun the pip command if you do some changes).

App Engine

You can setup an app in the [Google App Engine](#) by changing the `application` key in `app.yaml` to your application name. For using it you just need to do the API calls to `http://your-app-name.appspot.com`. It needs some external python modules in `lib`, you can download them by running `./devscripts/setup-gae.sh` (it requires [pip](#) and [virtualenv](#))

Heroku

You can also run the server on [Heroku](#), you just need to [setup and deploy the app](#) and you can make your API calls to `https://app-name.herokuapp.com/`.

CHAPTER 2

youtube-dl-server

Run the API server

- p** <port>, **--port** <port>
The port the server will use. The default port is 9191
- host** HOST
The host the server will use. The default host is localhost
- number-processes** NUMBER_PROCESSES
The number of processes the server will use. The default is: 5
- h, --help**
Display the help text
- version**
Print the version of the server

API methods

GET /api/info

Get the video information

Query Parameters

- **url** – The video url
- **flatten** (*boolean*) – If `True` return a list of dictionaries in the `videos` field. Otherwise a single dictionary will be returned in the `info` field.

Changed in version 0.2: The default value is `False`.

Deprecated since version 0.2: This parameter will be removed in a future version, you'll have to implement this functionality in your client.

- ***** – A whitelist of extra parameters are passed directly to the `YoutubeDL` object. Currently it supports: `format`, `playlist_items`, `rejecttitle`, `playlistend`, `matchtitle`, `playliststart`, `playlistreverse`, `subtitleslangs`, `subtitlesformat`, `allsubtitles`, `writeautomaticsub`, `writesubtitles`. See the [youtube-dl documentation](#) for more info

Response Headers

- **Content-Type** – `application/json`
- **Access-Control-Allow-Origin** – *

Status Codes

- **200 OK** – On success
- **400 Bad Request** – For invalid query parameters
- **500 Internal Server Error** – If the extraction fails

Example request

```
GET /api/info?url=http://www.ted.com/talks/dan_dennett_on_our_consciousness.html&
↳flatten=False HTTP/1.1
```

Example response

```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
Content-Type: application/json

{
  "info": {
    "description": "Philosopher Dan Dennett makes a compelling argument that not_
↳only don't we understand our own consciousness, but that half the time our_
↳brains are actively fooling us.",
    "display_id": "102",
    "ext": "mp4",
    "extractor": "TED",
    "extractor_key": "TED",
    "format": "high",
    "format_id": "high",
    "formats": [
      {
        "ext": "mp4",
        "format": "low",
        "format_id": "low",
        "preference": 1,
        "url": "...",
      },
      {
        "ext": "mp4",
        "format": "medium",
        "format_id": "medium",
        "preference": 2,
        "url": "...",
      },
      {
        "ext": "mp4",
        "format": "high",
        "format_id": "high",
        "preference": 3,
        "url": "...",
      }
    ],
    "id": "102",
    "playlist": null,
    "playlist_index": null,
    "preference": 3,
    "subtitles": null,
    "thumbnail": "http://images.ted.com/images/ted/488_480x360.jpg",
    "title": "The illusion of consciousness",
    "uploader": "Dan Dennett",
    "url": "...",
    "webpage_url": "http://www.ted.com/talks/dan_dennett_on_our_consciousness.html
↳",
    "webpage_url_basename": "dan_dennett_on_our_consciousness.html"
  },
  "url": "http://www.ted.com/talks/dan_dennett_on_our_consciousness.html"
}
```

Example request

```
GET /api/info?url=http://www.ted.com/talks/dan_dennett_on_our_consciousness.html&
↳flatten=True HTTP/1.1
```

Example response

```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
Content-Type: application/json

{
  "url": "http://www.ted.com/talks/dan_dennett_on_our_consciousness.html",
  "videos": [
    {
      "description": "Philosopher Dan Dennett makes a compelling argument that
↳not only don't we understand our own consciousness, but that half the time our
↳brains are actively fooling us.",
      "display_id": "102",
      "ext": "mp4",
      "extractor": "TED",
      "extractor_key": "TED",
      "format": "high",
      "format_id": "high",
      "formats": [
        {
          "ext": "mp4",
          "format": "low",
          "format_id": "low",
          "preference": 1,
          "url": "...",
        },
        {
          "ext": "mp4",
          "format": "medium",
          "format_id": "medium",
          "preference": 2,
          "url": "...",
        },
        {
          "ext": "mp4",
          "format": "high",
          "format_id": "high",
          "preference": 3,
          "url": "...",
        }
      ],
      "id": "102",
      "playlist": null,
      "playlist_index": null,
      "preference": 3,
      "subtitles": null,
      "thumbnail": "http://images.ted.com/images/ted/488_480x360.jpg",
      "title": "The illusion of consciousness",
      "uploader": "Dan Dennett",
      "url": "...",
      "webpage_url": "http://www.ted.com/talks/dan_dennett_on_our_consciousness.
↳html",
      "webpage_url_basename": "dan_dennett_on_our_consciousness.html"
    }
  ]
}
```

```
}  
]  
}
```

GET /api/play

Extract the info and redirect to the URL of the first video found for the requested URL. Useful for media players that accept HTTP URLs. Accepts the same parameters as *GET /api/info*.

Status Codes

- 302 Found – On success
- 400 Bad Request – For invalid query parameters
- 500 Internal Server Error – If the extraction fails

New in version 0.3: Added endpoint.

Example request

```
GET /api/play?url=http://www.ted.com/talks/dan_dennett_on_our_consciousness.html_  
↪ HTTP/1.1
```

Example response

```
HTTP/1.0 302 FOUND  
Content-Type: text/html; charset=utf-8  
Location: http://download.ted.com/talks/DanDennett_2003-1500k.mp4?dnt
```

Example request

```
GET /api/play?url=http://www.ted.com/talks/dan_dennett_on_our_consciousness.html&  
↪ format=bestaudio HTTP/1.1
```

Example response

```
HTTP/1.0 302 FOUND  
Content-Type: text/html; charset=utf-8  
Location: https://hls.ted.com/videos/DanDennett_2003/audio/600k.m3u8?  
↪ uniqueId=5ed2e870
```

GET /api/extractors

Get the available extractors

Response Headers

- Content-Type – application/json
- Access-Control-Allow-Origin – *

Status Codes

- 200 OK – On success

Example request

```
GET /api/extractors HTTP/1.1
```

Example response

```

HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
Content-Type: application/json

{
  "extractors": [
    {
      "name": "vimeo",
      "working": true
    },
    {
      "name": "TED",
      "working": true
    },
    ...
  ]
}

```

GET /api/version

Get the youtube-dl and youtube-dl-api-server version

Response Headers

- **Content-Type** – application/json
- **Access-Control-Allow-Origin** – *

Status Codes

- **200 OK** – On success

New in version 0.3: Added endpoint.

Example request

```
GET /api/version HTTP/1.1
```

Example response

```

HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
Content-Type: application/json

{
  "youtube-dl": "2016.04.19",
  "youtube-dl-api-server": "0.2"
}

```

Test server

You can try the API by doing requests to `http://youtube-dl.appspot.com`.

youtube-dl-api-server Changelog

Version 0.3

- Add `/api/play` endpoint. (Adapted from a patch by eadmaster in [PR #46](#))
- Add `/api/version` endpoint. (Added by Carlos Ramos in [PR #43](#))
- Pass the `format` parameter to the YoutubeDL object.
- Add `--number-processes` option for choosing the number of processes the server will use, defaults to 5. (Added by Carlos Ramos in [PR #42](#))
- Provide a Blueprint for the API.

Version 0.2

- `/api/` no longer redirects to `/api/info`.
- Change default value of the `flatten` parameter to `False` and deprecate it.
- Add `--host` option for choosing the server address. (Added by Alexandr Korsak in [PR #26](#))
- Add instructions and sample file for deploying to Heroku. (Thanks to Ronald Ip, see [PR #28](#))
- Force the `youtube-dl` `format` parameter to `best`. Since version `2015.04.26` it uses `bestvideo+bestaudio/best` if `ffmpeg/libav` is installed.
- Allow passing some extra parameters directly to the YoutubeDL object.

Version 0.1.2

- Fix disabling the `youtube-dl`'s cache in latest versions.

Version 0.1.1

- Fix access to the version string of youtube-dl in the last versions.

Version 0.1

- The endpoint for getting the video information is now `/api/info`. `/api/` redirects now to it, but it's deprecated and will be removed in a future version.
- The server can be run now with `python -m youtube_dl_server`.
- Added `flatten` option to `/api/info` to select between a single dictionary result or a list of videos, it's `True` by default.

Authors

youtube-dl-api-server is written and maintained by Jaime Marquínez Ferrándiz.

License

youtube-dl-api-server and its documentation is released to the public domain:

This is free and unencumbered software released into the public domain.

Anyone is free to copy, modify, publish, use, compile, sell, or distribute this software, either in source code form or as a compiled binary, for any purpose, commercial or non-commercial, and by any means.

In jurisdictions that recognize copyright laws, the author or authors of this software dedicate any and all copyright interest in the software to the public domain. We make this dedication for the benefit of the public at large and to the detriment of our heirs and successors. We intend this dedication to be an overt act of relinquishment in perpetuity of all present and future rights to this software under copyright law.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For more information, please refer to <<http://unlicense.org/>>



CHAPTER 6

Indices and tables

- `genindex`
- `modindex`
- `search`

HTTP Routing Table

/api

GET /api/extractors, 10
GET /api/info, 7
GET /api/play, 10
GET /api/version, 11

Symbols

- host HOST
 - youtube-dl-server command line option, 5
- number-processes NUMBER_PROCESSES
 - youtube-dl-server command line option, 5
- version
 - youtube-dl-server command line option, 5
- h , -help
 - youtube-dl-server command line option, 5
- p <port>, -port <port>
 - youtube-dl-server command line option, 5

Y

- youtube-dl-server command line option
 - host HOST, 5
 - number-processes NUMBER_PROCESSES, 5
 - version, 5
 - h , -help, 5
 - p <port>, -port <port>, 5