Stupeflix Tasks API Documentation

Release 0.1

Luper Rouch

November 17, 2014

Contents

The REST API is the interface to access the Stupeflix tasks system.

The API is versioned. You should always use the newest version of the API, but older versions will never be removed.

The base URL of the tasks system is `https://dragon.stupeflix.com/`. So for example the full URL for the method `/foo/bar` in the `v1/api` API is `https://dragon.stupeflix.com/v1/foo/bar`.

Contents:

CHAPTER 1

API Reference

1.1 General notes

All requests are done over SSL.

All strings must be UTF-8 encoded.

POST requests parameters are passed in JSON-encoded bodies.

All dates are in ISO8601 format.

1.2 Authentication

Requests that create tasks such as `/v1/create` require authentication. There are two methods to authenticate requests:

1.2.1 Secret key

You can add your secret key to requests parameters, here for example for the `/v1/create` method:

```
{
    "secret": "123456",
    "tasks": {
        "task_name": "image.info",
        "url": "http://files.com/image.jpg"
    }
}
```

The secret key can also be passed via the `Authorization` header. The key should be prefixed by the string `Secret`, with whitespace separating the two strings:

```
Authorization: Secret 123456
```

This kind of authentication should only be used for server to server requests.

1.2.2 Api key + referrer

As for *Secret key* authentication, the api key can be passed in the request parameters or the `Authorization` header:

- in request parameters:

```
{
    "api_key": "654321",
    "tasks": {
        "task_name": "image.info",
        "url": "http://files.com/image.jpg"
    }
}
```

- in the `Authorization` header:

```
Authorization: Api-Key 654321
```

The `Referrer` header of the request must also be in your account's whitelist. This kind of authentication is what you should use in your javascript code.

1.3 HTTP Status codes

200 Operation was successful.

400 Invalid request parameters. The response body contains a description of the errors, for example if you forgot the `tasks` parameter in a `/v1/create` request:

```
{
    "status": "error",
    "errors": [
        {
            "name": "tasks",
            "location": "body",
            "description": "tasks is missing"
        }
    ]
}
```

401 Invalid credentials, or account limits reached.

1.4 Tasks definitions

Task are defined by objects with at least a `task_name` key. Other keys contain the task parameters. Here is an example of a `image.info` task definition, which takes an `url` parameter:

```
{"task_name": "image.info", "url": "http://files.com/image.jpg"}
```

1.5 Tasks statuses format

Tasks statuses are objects of the form:

```
{
    "status": "executing",
    "key": "5OYA5JQVFIAHYOMLQG5QV3U33M",
    "progress": 90,
    "events": {
        "started": "2013-04-03T15:47:27.707526+00:00",
        "queued": "2013-04-03T15:47:27.703674+00:00"
```

```
    }
}
```

Here are the descriptions of the keys:

- `status`: the current step of the task in the execution pipeline, one of "queued", "executing, "success" or "error"
- `key`: the server-side key used to identify the task
- `progress`: a value representing task progress; its type depends on the task and could be anything that is JSON-encodable
- **`events`: an object containing chronological events of the task:**
  - `queued`: date at which the task was queued
  - `started`: date at which the task has been attributed to a worker
  - `completed`: completion date of the task

Tasks results (with a `status` "success" or "error") also contain an additional `result` key, containing the task result for successful tasks, or the details of the error.

Error details can be a simple string, or an object for parameters or result validation errors. Here is an example error status where a required field named "url" was omitted:

```
{
    "status": "error",
    "progress": null,
    "events": {
        "completed": "2013-09-20T12:56:49.385937+00:00",
        "queued": "2013-09-20T12:56:49.369911+00:00"
    },
    "key": "QJZTXA3LNZKQ6X4RPGQ5EHRSMI",
    "result": {
        "parameters": {
            "url": [
                "this field is required"
            ]
        }
    }
}
```

1.6 Tasks results

Warning: Tasks results are ephemeral!

Once created tasks statuses and tasks output files will be available for about one week. After that period, accessing a status or an output file will return a 404.

Tasks output files are designed to be downloaded and then served by your means.

1.7 API Methods

`POST` **`/v1/create`**
Queue one or more tasks and return a list of tasks status.

Here is an example request creating two tasks:

```
{
    "tasks": [
        {"task_name": "image.info", "url": "http://files.com/image.jpg"},
        {"task_name": "image.thumb", "url": "http://files.com/image.jpg"}
    ]
}
```

Query params **tasks** – a list containing the definitions of the tasks to execute. The method also accepts a single task definition for convenience.

Opt. params **block** – a boolean indicating if the call should return immediately with the current status of the tasks, or wait for all tasks to complete and return their final status.

Response a list of task statuses:

```
[
    {
        "status": "queued",
        "events": {"queued": "2013-04-03T15:47:27.703674+00:00"},
        "key": "6GRQ3H5EHU7GXUTIOSS2GUDPGQ"
    },
    {
        "status": "queued",
        "events": {"queued": "2013-04-03T15:47:27.703717+00:00"},
        "key": "5OYA5JQVFIAHYOMLQG5QV3U33M"
    }
]
```

GET **/v1/create_file/*{filename}***

This is the GET version of `tasks_file_post`, allowing to create a task, and redirect to one of its output files by using GET semantics.

It can be useful for cases where you want to use directly the result of a task but can't issue a POST. For example you could create a thumbnail directly in an image tag:

```
<img src="https://dragon.stupeflix.com/v1/create_file/cat.jpg?task_name=image.thumb&url=http://f
```

Path arguments **filename** – used to select the desired file, for tasks that output multiple files. Can be omited for tasks that output a single file.

Query params

- **task_name** – task name.
- ***** – remaining querystring parameters are the parameters of the task.

Response returns the same responses as `create_file_post`.

GET **/v1/status**

Query the status of one or more tasks.

Example request:

```
https://dragon.stupeflix.com/v1/status?tasks=6GRQ3H5EHU7GXUTIOSS2GUDPGQ&tasks=5OYA5JQVFIAHYOMLQG
```

Query params

- **tasks** – one or more tasks keys.

- **block** – a boolean indicating if the call should return immediately with the current status of the task, or wait for all tasks to complete and return their final status.

Response a list of task statuses, see `create` for a response example.

`POST` **`/v1/status`**
Same as `status` but using POST semantics. Usefull when there are too much tasks to query and the querystring size limit is reached.

`GET` **`/v1/status_stream`**
Get status streams of one or more tasks.

Example request:

```
https://dragon.stupeflix.com/v1/stream?tasks=6GRQ3H5EHU7GXUTIOSS2GUDPGQ&tasks=5OYA5JQVFIAHYOMLQG
```

Query params **tasks** – one or more tasks keys.

Response a stream of status updates. See `create_stream` for a description of the output.

`POST` **`/v1/status_stream`**
Same as `status_stream` but using POST semantics. Usefull when there are too much tasks to query and the querystring size limit is reached.

`GET` **`/v1/file/`*`{filename}`***
Wait for an existing task to complete and redirect to its output.

Example request:

```
https://dragon.stupeflix.com/v1/file/cat.jpg?task=6GRQ3H5EHU7GXUTIOSS2GUDPGQ
```

Path arguments **filename** – used to select the desired file, for tasks that output multiple files. Can be omited for tasks that output a single file.

Query params **task** – the task key.

Response returns the same responses as `create_file_post`.

CHAPTER 2

Tasks Reference (v1)

2.1 audio.beats

Find beats in an audio file.

Parameters

- **url** (*string*) – URL of the audio file

Output Values

- **beats** (*object*) – An array containing the timestamps of the detected beats, in seconds
- **duration** (*integer*) – The processed audio file duration in seconds

2.2 audio.convert

Transcode audio file (mp3, vorbis), and return audio duration.

Parameters

- **url** (*string*) – URL of the audio file to be converted.
- **codec** (*string*) – Desired codec for the output file. *(choices:* `'mp3'`, `'vorbis'` *) (default:* `u'mp3'` *)*

Output Values

- **duration** (*float*) – Duration of the audio file in seconds, rounded to 1/100th second.
- **content_type** (*string*) – Output file content type.

Output Files

- **output** – URL of the output file.

2.3 audio.info

Return duration and codec of an audio file.

Parameters

- **url** (*string*) – URL of the audio file to be scanned.

Output Values

- **duration** (*float*) – Duration of the audio file in seconds, rounded to 1/100th second.
- **content_type** (*string*) – Content-type of the audio file.
- **codec** (*string*) – Codec of the audio file.

2.4 audio.tts

Create audio voice-over file using Text-to-Speech (US English, male/female voices), returns duration.

Parameters

- **text** (*string*) – Text string to be transformed into audio via speech synthesys.
- **voice** (*string*) – 1 male and 1 female US English voices are available. *(choices:* `'neospeech:julie','neospeech:paul'` *) (default:* `u'neospeech:julie'` *)*
- **codec** (*string*) – 1 male and 1 female US English voices are available. *(choices:* `'mp3'`, `'ogg'` *) (default:* `u'mp3'` *)*

Output Values

- **duration** (*float*) – Duration of the audio file in seconds, rounded to 1/100th second.
- **content_type** (*string*) – Content-type of the audio file.

Output Files

- **output** – URL of the output file.

2.5 audio.waveform

Create a waveform image from an audio file.

Parameters

- **url** (*string*) – URL of the audio file to be scanned.
- **width** (*integer*) – *(default:* `1024` *)*
- **height** (*integer*) – *(default:* `60` *)*
- **vmargin** (*integer*) – vertical margin *(default:* `0` *)*
- **fill** (*string*) – Color of the wave-form. *(default:* `u'#000000'` *)*
- **background** (*string*) – Color of the background. *(default:* `u'#FFFFFF'` *)*
- **start** (*float*) – seconds to start from. *(default:* `0.0` *)*
- **end** (*float*) –
- **format** (*string*) – *(choices:* `'png','jpeg'` *) (default:* `u'jpeg'` *)*

Output Values

- **duration** (*float*) – Duration of the audio file in seconds, rounded to 1/100th second.
- **width** (*integer*) –
- **height** (*integer*) –
- **content_type** (*string*) –

Output Files

- **output** – URL of the output file.

2.6 html.scrape

Scrape html webpage to return videos & images found

Parameters

- **url** (*string*) – URL of the html page

Output Values

- **hits** (*object*) –
- **page_title** (*string*) –

2.7 image.face

Return an array of positions of detected faces, with type and confidence.

Parameters

- **url** (*string*) – URL of the analyzed image.

Output Values

- **faces** (*object*) – An array containing salient points coordinates.

2.8 image.gif

Create an animated GIF from a list of images.

Parameters

- **images** (*list of strings*) – The list of image URLs that will be used to create the animated GIF.
- **loop** (*integer*) – The number of loops of the GIF, 0 means to loop forever. *(default:* `0` *)*
- **frame_duration** (*float*) – The duration in seconds during which each image will be shown when the GIF is playing, rounded to 1/100th of a second. *(default:* `0.1` *)*
- **width** (*integer*) – The pixel width of the output GIF. Leave empty to use source images width.

- **height** (*integer*) – The pixel height of the output GIF. Leave empty to use source images height.

Output Files

- **output** – The URL of the output GIF.

2.9 image.info

Return image file information.

Parameters

- **url** (*string*) – URL of the image file to be scanned.

Output Values

- **content_type** (*string*) – Content-Type of the image file.
- **type** (*string*) – Type of the file.
- **width** (*integer*) –
- **height** (*integer*) –
- **alpha** (*boolean*) –
- **rotation** (*float*) – The rotation that should be applied to the image to see it as it was shot, in degrees.
- **date_time** (*string*) –
- **flash** (*boolean*) –
- **focal_length** (*float*) –
- **iso_speed** (*float*) –
- **exposure_time** (*float*) –

2.10 image.saliency

Return an array of salient points coordinates within an image.

Parameters

- **url** (*string*) – URL of the analyzed image.

Output Values

- **points** (*object*) – An array containing salient points coordinates.

2.11 image.smartcrop

Return most interesting (entropy based), non-overlapping rectangles, for a given surface ratio, within an image.

Parameters

- **url** (*string*) – URL of the image file to be scanned.
- **aspect_ratio** (*float*) – *(default:* `1.7777777777777777` *)*
- **boxes_number** (*integer*) – *(default:* `10` *)*
- **step_ratio** (*float*) – *(default:* `0.03` *)*
- **diag_ratio** (*float*) – *(default:* `0.3` *)*
- **reverse** (*boolean*) – *(default:* `False` *)*

Output Values

- **points** (*object*) – the JSON dump of the result

2.12 image.thumb

Create a new image of custom dimensions and orientation from an original image.

Parameters

- **width** (*integer*) – Desired thumbnail width, in pixels.
- **height** (*integer*) – Desired thumbnail height, in pixels
- **crop** (*boolean*) – If crop is true, original image fills new image dimensions. If crop is false, original image fits new image dimensions. *(default:* `False` *)*
- **url** (*string*) – URL of the source image
- **rotation** (*integer*) – A counter clockwise rotation rotation to apply to the thumbnail, in degrees. *(choices:* `0`, `90`, `180`, `270` *) (default:* `0` *)*
- **poster** (*boolean*) – If true, a play icon is added in the center. *(default:* `False` *)*
- **format** (*string*) – The output format. *(choices:* `'jpeg'`, `'gif'`, `'png'` *) (default:* `u'jpeg'` *)*

Output Values

- **width** (*integer*) – thumbnail width
- **height** (*integer*) – thumbnail height
- **original_width** (*integer*) – original image width
- **original_height** (*integer*) – original height

Output Files

- **output** – URL of the thumbnail.

2.13 video.convert

Create transcoded video file with custom dimensions, and return its video.info output values.

Parameters

- **url** (*string*) – URL of the video file to convert.

- **width** (*integer*) –
- **height** (*integer*) –
- **crop** (*boolean*) – Allows croping the video to fit in the output size *(default:* `False` *)*
- **audio_codec** (*string*) – Desired audio audio. *(choices:* `'mp2'`, `'mp3'`, `'aac'`, `'wmav1'`, `'wmav2'` *) (default:* `u'aac'` *)*
- **video_codec** (*string*) – Desired video codec. *(choices:* `'h264'` *) (default:* `u'h264'` *)*
- **video_bitrate** (*integer*) – Desired video bitrate, in kbps. *(default:* `512` *)*
- **audio_bitrate** (*integer*) – Desired audio bitrate, in kbps. *(default:* `64` *)*
- **sample_rate** (*integer*) – Desired audio sample rate, in kHz. *(choices:* `22050`, `44100`, `48000` *) (default:* `48000` *)*

Output Values

- **content_type** (*string*) – Output file content type.
- **width** (*integer*) –
- **height** (*integer*) –
- **original_width** (*integer*) –
- **original_height** (*integer*) –
- **duration** (*float*) – Duration of the video file, in seconds.
- **frame_rate** (*float*) –
- **audio_codec** (*string*) –
- **video_codec** (*string*) –
- **alpha** (*boolean*) –
- **rotation** (*float*) – The counter clockwise rotation that should be applied to the video to see it as it was shot, in degrees.

Output Files

- **output** – URL of the converted file.

2.14 video.create

Create video file(s) from a XML definition and video profile(s).

Parameters

- **definition** (*string*) –
- **preview** (*boolean*) – *(default:* `True` *)*
- **export** (*boolean*) – *(default:* `True` *)*
- **profile** (*string*) – *(choices:* `'iphone-24p'`, `'dvd-pal-16-9'`, `'360p'`, `'360p-23-976-fps'`, `'480p-4-3-29-97-fps'`, `'dvd-ntsc-4-3-h'`, `'dvd-pal-4-3-h'`, `'360p-24-fps'`, `'360p-12-5-fps'`, `'1080p-24-fps'`, `'youtube-12-5fps'`, `'dvd-pal-4-3'`, `'480p-24-fps'`, `'iphone-slow'`, `'ntsc-wide-wmv'`, `'special'`, `'360p-11-988-fps'`, `'dvd-mpeg1-small'`,

'youtube-flv', '720p-12-fps', 'dvd-pal-16-9-h', 'youtube-slow', '720p-12-5-fps', 'wmv2', 'flash', 'flash-hq', 'mobile-small', 'youtube-5fps', 'flash-large-4-3', 'iphone', '720p-24-fps', 'iphone-flv', 'iphone-16-9-12fp', '1080p', 'wmv1', '240p-24-fps', 'iphone-16-9', 'quicktime', '720p-23-98-fps', 'th720p', '360p-29-97-fps', 'youtube-slow-flv', 'wmv2-large-4-3', 'dvd-mpeg1', 'ntsc-wide', 'flash-small', 'dvd-ntsc-16-9', '480p', 'dvd-ntsc-4-3', 'mobile', 'iphone-sslow', '720p', 'youtube', '720p-hq', 'square-400', 'dvd-ntsc-16-9-h', 'iphone-16-9-slow', 'cine-half-hd', 'flash-h264', '240p', 'quicktime-small', '720p-29-97-fps', '360p-12-fps', 'flash-med-16-9') *(default:* u'360p' *)*

- **thumbnail_time** (*float*) – *(default:* 1.0 *)*
- **url_callback** (*string*) –

Output Values

- **duration** (*float*) –
- **width** (*integer*) – video width
- **height** (*integer*) – video height

Output Files

- **preview** –
- **export** –
- **thumbnail** –

2.15 video.info

Return video file information.

Parameters

- **url** (*string*) – URL of the video file to be scanned.

Output Values

- **content_type** (*string*) – Mime-type of the video file.
- **width** (*integer*) – Video width, in pixels.
- **height** (*integer*) – Video height, in pixels.
- **duration** (*float*) – Video duration, in seconds.
- **frame_rate** (*float*) – Video frame rate, in frames per second.
- **alpha** (*boolean*) – A boolean indicating if the video has an alpha channel.
- **rotation** (*float*) – The rotation that should be applied to the video to see it as it was shot, in degrees.
- **audio_codec** (*string*) – Audio codec name.
- **video_codec** (*string*) – Video codec name.

2.16 video.reverse

Create a reversed video file with custom dimensions, and return its video.info output values.

Parameters

- **url** (*string*) – URL of the source video.
- **width** (*integer*) – Desired width of the reversed video. If left unspecified, keep the original width.
- **height** (*integer*) – Desired height of the reversed video. If left unspecified, keep the original height.
- **audio_codec** (*string*) – Desired audio codec. *(choices:* `'mp2'`, `'mp3'`, `'aac'`, `'wmav1'`, `'wmav2'` *)* *(default:* `u'aac'` *)*
- **video_codec** (*string*) – Desired video codec. *(choices:* `'h264'` *)* *(default:* `u'h264'` *)*
- **video_bitrate** (*integer*) – Desired video bitrate, in kbps. *(default:* `512` *)*
- **audio_bitrate** (*integer*) – Desired audio bitrate, in kbps. *(default:* `64` *)*
- **sample_rate** (*integer*) – Desired audio sample rate, in kHz. *(choices:* `22050`, `44100`, `48000` *)* *(default:* `48000` *)*

Output Values

- **duration** (*float*) – Duration of the input video .file, in pixels

Output Files

- **output** – URL of the reversed video file.

2.17 video.strip

Create a film strip image of custom dimensions showing stitched frames of a video, return video.info output values for original video.

Parameters

- **url** (*string*) – URL of the source video.
- **width** (*integer*) – Pixel width of each frame stitched into film strip.
- **height** (*integer*) – Pixel height of each frame stitched into film strip.
- **crop** (*boolean*) – If false, video frames fit each strip section. If true, video frames fill each strip section, aligning centers. *(default:* `False` *)*
- **wrap** (*integer*) – Number of video frames that can be stitched horizontally before stitching starts onto a new line. Use it to create a two dimensional film strip, with count = int * wrap. If left unspecified, all frames are stitched on a single line.
- **start** (*float*) – Time of first frame extracted from video - by default first frame of video. *(default:* `0.0` *)*
- **end** (*float*) – Time of last frame extracted from video - by default last frame of video.
- **count** (*integer*) – Number of frames extracted from video, at equal time intervals between start and end times. *(default:* `10` *)*

- **format** (*string*) – Output image file format *(choices:* `'jpeg'`, `'png'` *)* *(default:* `u'jpeg'` *)*

Output Values

- **count** (*integer*) – Actual number of frames in the output.
- **width** (*integer*) – Width of the output image in pixels.
- **height** (*integer*) – Height of the output image in pixels.
- **original_width** (*integer*) – Width of the input video file, in pixels.
- **original_height** (*integer*) – Width of the input video file, in pixels.
- **duration** (*float*) – Duration of the input video .file, in pixels
- **frame_rate** (*float*) – Frame rate of the input video file, in frames per second.
- **content_type** (*string*) – Mime-type of the output image.

Output Files

- **output** – URL of the output image.

2.18 video.thumb

Create an image of custom dimensions extracted at a specified time in a video.

Parameters

- **url** (*string*) – URL of the source video.
- **width** (*integer*) – Width of output image file, in pixels. The default is to use the original video width.
- **height** (*integer*) – Height of output image file, in pixels. The default is to use the original video height.
- **crop** (*boolean*) – If false, video frame fits output image. If true, video frame fills output image. *(default:* `False` *)*
- **time** (*float*) – Timestamp of the video frame to extract, in seconds. *(default:* `0.0` *)*
- **format** (*string*) – Output image file format. *(choices:* `'jpeg'`, `'png'` *)* *(default:* `u'jpeg'` *)*
- **poster** (*boolean*) – If true, a play icon is added in the center. *(default:* `False` *)*

Output Values

- **width** (*integer*) – Width of the output image in pixels.
- **height** (*integer*) – Height of the output image in pixels.
- **original_width** (*integer*) – Width of the input video file.
- **original_height** (*integer*) – Width of the input video file.
- **duration** (*float*) – Duration of the input video file, in seconds.
- **content_type** (*string*) – Mime-type of the output image.

Output Files

- **output** – URL of the output image.

2.19 video.upload.fb

Upload a video to Facebook.

Parameters

- **url** (*string*) – URL of the source video.
- **api_key** (*string*) – Facebook API key.
- **app_secret** (*string*) – Facebook app secret.
- **access_token** (*string*) – Target user's access token.
- **title** (*string*) – Video title.
- **description** (*string*) – Video description.

Output Values

- **duration** (*float*) – Duration of the input video file, in seconds.

Output Files

- **output** – URL of the uploaded video on Facebook.

2.20 video.upload.vimeo

Upload a video from user url on Vimeo. Register your app to get a consumer key and secret. Then retrieve an access token key and a secret following these instructions on Oauth for the Vimeo API.

Parameters

- **url_callback** (*string*) –
- **url** (*string*) – Video url to upload
- **title** (*string*) – Video title
- **description** (*string*) – Video description
- **consumer_key** (*string*) – Application consumer key
- **consumer_secret** (*string*) – Application consumer secret
- **access_token_key** (*string*) – User access token key
- **access_token_secret** (*string*) – User access token secret

Output Values

- **free_space** (*integer*) –
- **uploaded_file_size** (*integer*) –
- **output** (*string*) – URL of the uploaded video on Vimeo.

2.21 video.upload.youtube

Upload a video to Youtube using the version 3 of the API with OAuth2 Bearer authentication. Register your app and retrieve an access token following these instructions.

Otherwise, you can also get a token with us from there

Parameters

- **url** (*string*) – URL of the source video.
- **access_token** (*string*) – Target user's access token with upload authorization.
- **developer_key** (*string*) – Youtube developer key of a registered app.
- **title** (*string*) – Video title.
- **description** (*string*) – Video description.
- **tags** (*list of strings*) – *(default:* `[]` *)*
- **category_id** (*integer*) – Video category ID number.The default value is 22, which refers to the People & Blogs category.
- **privacy_status** (*string*) – Privacy status of the video. *(choices:* `'public'`, `'private'`, `'unlisted'` *) (default:* `u'public'` *)*

Output Values

- **output** (*string*) – URL of the uploaded video on Youtube.
- **duration** (*float*) – Duration of the input video file, in seconds.

CHAPTER 3

Indices and tables

- *genindex*
- *search*

a

h

i

v

Symbols

A

G

H

I

P

V