
Stupeflix Factory Documentation

Release 2

Stupeflix

February 22, 2016

1	Getting started	3
1.1	Editing sessions	3
1.2	Configure a session	3
1.3	Display the editing interface	5
1.4	Get the produced video info	5
1.5	Get the status of a session	5
2	How to implement a Custom Media Library Server	7
2.1	Requests	7
2.2	Responses	8

Stupeflix Factory allows you to integrate a feature complete [Stupeflix Editor](#) into your application flow.

Getting started

A Stupeflix Factory demo is available at <http://studio.stupeflix.com/factory/demo/>.

1.1 Editing sessions

Stupeflix Factory works with editing sessions. One editing session ends with one produced video.

Every editing session goes like this:

1. Configure/Create a session
2. Display the editing interface
3. Get the produced video info

1.2 Configure a session

POST your configuration to the following endpoint to create a session:

```
https://studio.stupeflix.com/factory/v3/
```

1.2.1 Configuration parameters

Stupeflix Factory works like basic html forms with `action`, `method` and `target` parameters defining how the factory will call you server back.

action The URL of the program that will process the information submitted by the factory. If this value is not specified the factory will submit to a debug page.

method The HTTP method to use to submit the factory information to the `action` URL. Possible values are `get` (default) or `post`.

target

A name or keyword indicating where to display the response returned by the `action` URL.

- `_self`: Load the response into the same frame as the factory.
- `_blank`: Load the response into a new unnamed window.
- `_parent`: Load the response into the factory parent frame. If there is no parent, this option behaves the same way as `_self`.

- `_top` (default): Load the response into the factory top-level parent frame. If there is no parent, this option behaves the same way as `_self`.

video_name The name you want to give to your user's video (your user will be able to change it).

export_profile The video profile to use to export the video. It defines width, height, bitrate and codecs of the video to be produced. See [Stupeflix API supported formats](#) for a list of the available profiles. Default is `youtube`.

xml_skin The Stupeflix XML you want to wrap the user's video into. Useful to add pre-roll, post-roll or watermarking to the produced videos. The provided XML must contain a `{{user_movie}}` placeholder to show the user's video position. [XML example](#).

allow_preview Defines if the factory allows the user to preview her video. Possible values are `true` (default) or `false`.

custom_css_url The URL of an extra css file to include into the factory html pages. This url must be served over https if you choose to display your factory using https.

export_label The label to display for the export button, default is `Export`.

confirm_export_label If defined, asks the user to confirm her decision to export and exit the editor by displaying `confirm_export_label` in a confirm box. Undefined by default.

rendering_label The label to display on the video rendering screen (progress bar), default is `Please wait while Stupeflix.com creates your video...`

store_type

Either **volatile** or **persistent**. Defines how this session produced video will be stored. Default is **volatile**.

- Volatile files are stored for a week on Stupeflix servers.
- **Persistent files** are hosted on Amazon S3 and delivered through Amazon CloudFront CDN. Additional costs apply to persistent files, [please read our pricing](#).

secret Your Stupeflix API key secret is needed to authenticate your configuration call. Manage your Stupeflix API keys on the [Stupeflix Developers website](#).

1.2.2 Advanced configuration parameters

video_id Re-configure an existing non-complete editing session. Supports all standard configuration parameters except for `export_profile`. `video_id` must be a valid non-complete editing session identifier.

remix Clone an existing editing session. `remix` must be a valid editing session identifier.

definition Use the `definition` parameter to prepopulate the editor. The `definition` value must be a valid json project definition.

custom_media_library_url

The URL of a custom media library server. See [How to implement a Custom Media Library Server](#).

If configured Stupeflix Factory will display the custom media library option alongside the others media importers.

custom_media_library_name The name to display for the custom media library option.

1.2.3 Response

A successfull configuration call returns a JSON response like this:


```
{
  "video_id": "12345"
  "video_name": "My Stupeflix Video",
  "status": "editing",
  "edit_url": "https://studio.stupeflix.com/factory/v3/edit/?video_id=12345"
  "created_at": "2013-12-17"
}
```

video_id This editing session unique identifier. As a session ends being a video, the session parameter is called `video_id`. That's the identifier you have to associate with your users if you want to keep track of which videos were created by which users.

video_name The name of the created video.

status The status of the editing session, possible values are `editing`, `rendering` or `complete`.

edit_url This editing session url, to display in an `iframe`.

created_at The iso format utc datetime when the editing session was created.

1.3 Display the editing interface

Once you get an `edit_url`, you have to display its content to your user. You can either do it by redirecting your user's browser to this url or by setting this url as an `iframe` src:

```
<iframe src="https://studio.stupeflix.com/factory/v3/edit/?video_id=12345"
  width="960" height="600" scrolling="no" frameborder="no"></iframe>
```

1.4 Get the produced video info

When your user's video is ready, Stupeflix Factory will call your server back, respecting your `action`, `method` and `target` configuration with the following data:

video_id This editing session unique identifier.

video_name The name your user gave to her video.

video_url The URL of the exported video file.

thumb_url An URL pointing to a thumbnail of the exported video.

hres The horizontal (x) resolution of the exported video and thumbnail.

vres The vertical (y) resolution of the exported video and thumbnail.

1.5 Get the status of a session

Session' statuses are available anytime at the following endpoint, passing in `video_id` as your session identifier:

```
https://studio.stupeflix.com/factory/v3/status/?video_id=12345
```

How to implement a Custom Media Library Server

2.1 Requests

Stupeflix Factory will query the configured `custom_media_library_url` each time an user wants to access the custom library content.

2.1.1 Root level content

To get a 50 items list starting at the item n°0, Stupeflix Factory will issue a GET request to:

```
http://custom_media_library_url?method=getList&offset=0&limit=50
```

GET parameters:

name	value
method	getList
offset	0
limit	50

to get the next 50 items the request will be:

```
http://custom_media_library_url?method=getList&offset=50&limit=50
```

GET parameters:

name	value
method	getList
offset	50
limit	50

2.1.2 Folder content

To get the 50 first items of the folder 562436708, Stupeflix Factory will issue a GET request to:

```
http://custom_media_library_url?method=getList&id=562436708&offset=50&limit=50
```

GET parameters:

name	value
method	getList
id	562436708
offset	0
limit	50

2.2 Responses

custom_media_library_url server responses are expected in JSON format (<http://www.json.org/>) with an application/javascript Content-Type.

```
{
  /* offset & limit values used for this response */
  "offset": 0,
  "limit": 50,

  /* Items to display (array)
     supported types are folder, image, video */

  "items": [

    {
      "type": "folder",
      "id": "562436708",      /* Unique folder identifier on your server.
                             Stupeflix Factory will issue a GET request to:
                             http://custom_media_library_url?method=getList&id=562436708&offset=0&
                             to get this folder content */

      "title": "Folder A",  /* Name to display */

      "thumb_url": "http://whatever/thumbnail/url.jpg", /* Url of the thumbnail to display */

      /* Optional properties */
      "num_entries": 10 /* Number of entries in this folder */
    },

    {
      "type": "image",
      "title": "Image 1",
      "file_url": "http://image-1/url.jpg",
      "thumb_url": "http://image-1/thumbnail/url.jpg",

      /* Optional properties */
      "height": 800,    /* Height of the image in pixels
                        // Width of the image in pixels
      "width": 600
    },

    {
      "type": "video",
      "title": "Video 1",
      "file_url": "http://video-1/url.mp4",
      "thumb_url": "http://video-1/thumbnail/url.jpg",

      /* Optional properties */
      "height": 640,    /* Height of the video in pixels
                        // Width of the video in pixels
      "width": 480,
      "duration": 120  /* Duration of the video in seconds
    }

  ],

  /* Does the server have more items in this list ?
     If has_more is true, Stupeflix Factory will issue a new GET request with an incremented offset p
  "has_more": false
}
```

```
}
```