

HD Cloud API Overview

January 22, 2010

support@hdcloud.com

Table of Contents

API Summary	1
About	1
Authentication	1
Formats	1
Params	1
Job Resources	2
All	2
Current	2
Completed	3
Failed	3
Create	4
Show	6
Frames	7
Destroy Bulk	8
Store Resources	8
All	8
Show	8
Profile Resources	9
All	9
Show	10

API Summary

About

The API is a REST-like interface heavily oriented around the core resources of HD Cloud: Jobs, Encoding Profiles, and Stores.

Authentication

Currently, HTTP Basic Auth is supported using your web account credentials.

If you have developer tokens you can use the token's key and passphrase as you would your username and password to authenticate. To create a token please visit your Account page.

Formats

You can request resources as XML or JSON.

- <http://hdcloud.com/api/v1/jobs.xml>
- <http://hdcloud.com/api/v1/jobs/current.json>

You may specify the format as part of a query string, as well.

- <http://hdcloud.com/api/v1/jobs?format=xml>

In the absence of a :format, proper HTTP Accept headers will do.

XML text/xml, application/x-xml

JSON text/x-json, application/jsonrequest

Params

Format is always required. See "Formats" for more about how you can specify the format param. If you specify no format or an invalid format, an HTTP 406 Not Acceptable response is sent.

Page is required for paginated actions, but will default to a value of 1 if not specified by you.

In general, params noted for an action are required unless specified as optional.

Param names like **files[]** imply an array which should be noted in params like so: **files[]=foo.mpg&files[]=bar.avi**. See the Jobs create action for examples of this.

Params like **job[source_id]** imply a hash or dictionary like structure which should be noted like so: **job[source_id]=1&job[destination_id]=1**. See the Jobs create action for examples of this.

Pagination

When an action specifies that it is paginated, this means that it will return a subset of all items from a collection. You can page through subsets using a **page** parameter.

- <http://hdcloud.com/api/v1/jobs?format=json&page=2>

Job Resources

All

GET <http://hdcloud.com/api/v1/jobs>

Get all jobs you have ever created. Paginated.

EXAMPLE RESPONSES

XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<jobs type="array">
  <job>
    <id>1</id>
    <current_step>encoding</current_step>
    <current_progress>42</current_progress>
    <current_status>Encoding: Pass 1</current_status>
    <source_filename>beer-drinking-pig.mpg</source_filename>
    <status_url>http://example.com/api/v1/jobs/1</status_url>
  </job>
</jobs>
```

JSON:

```
{{"job":
  {"source_filename": "beer-drinking-pig.mpg",
   "current_step": "encoding",
   "status_url": "http://example.com/api/v1/jobs/1",
   "id": 1,
   "current_status": "Encoding: Pass 1",
   "current_progress": 42}}
```

Current

GET <http://hdcloud.com/api/v1/jobs/current>

Get all jobs in progress. Paginated.

EXAMPLE RESPONSES

XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<jobs type="array">
  <job>
    <id>1</id>
    <current_step>encoding</current_step>
    <current_progress>42</current_progress>
    <current_status>Encoding: Pass 1</current_status>
    <source_filename>beer-drinking-pig.mpg</source_filename>
    <status_url>http://example.com/api/v1/jobs/1</status_url>
  </job>
</jobs>
```

JSON:

```
{["job":
  {"source_filename": "beer-drinking-pig.mpg",
   "current_step": "encoding",
   "status_url": "http://example.com/api/v1/jobs/1",
   "id": 1,
   "current_status": "Encoding: Pass 1",
   "current_progress": 42}]}
```

Completed

GET <http://hdcloud.com/api/v1/jobs/completed>

Get all completed jobs. Paginated.

EXAMPLE RESPONSES

XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<jobs type="array">
  <job>
    <id>1</id>
    <current_step></current_step>
    <current_progress></current_progress>
    <current_status>completed</current_status>
    <source_filename>beer-drinking-pig.mpg</source_filename>
    <status_url>http://example.com/api/v1/jobs/1</status_url>

    <encoded_filename>beer-drinking-pig_1_800_1920x1080.mp4</encoded_filename>
    <complete>true</complete>
  </job>
</jobs>
```

JSON:

```
{["job":
  {"source_filename": "beer-drinking-pig.mpg",
   "current_step": nil,
   "status_url": "http://example.com/api/v1/jobs/1",
   "id": 1,
   "complete?": true,
   "current_status": "completed",
   "encoded_filename": "beer-drinking-pig_1_800_1920x1080.mp4",
   "current_progress": nil}]}
```

Failed

GET <http://hdcloud.com/api/v1/jobs/failed>

Get all failed jobs. Paginated.

EXAMPLE RESPONSES

XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<jobs type="array">
  <job>
    <id>1</id>
    <current_step>failed</current_step>
    <current_progress></current_progress>
    <current_status>failed</current_status>
    <source_filename>beer-drinking-pig.mpg</source_filename>
    <status_url>http://example.com/api/v1/jobs/1</status_url>
    <failed>true</failed>
    <failure_message>An unexpected application error occurred. Please contact
support.</failure_message>
    <failure_code>500</failure_code>
  </job>
</jobs>
```

JSON:

```
[{"job":
  {"source_filename": "beer-drinking-pig.mpg",
    "current_step": "failed",
    "status_url": "http://example.com/api/v1/jobs/1",
    "failed?": true,
    "id": 1,
    "current_status": "failed",
    "failure_code": 500,
    "failure_message":
      "An unexpected application error occurred. Please contact support.",
    "current_progress": nil}}
```

Create

POST <http://hdcloud.com/api/v1/jobs>

Create one or more new encoding jobs.

PARAMETERS

job[source_id]

An integer referring to a Store ID.

job[destination_id]

An integer referring to a Store ID.

job[priority]

An integer from 1 to 10 which determines the order in which your jobs are processed. Higher numbers are higher priority.

It is recommended that you aim for a mid-level priority in general so that you have availability to run high priority jobs when you really need it. e.g. user generated content that needs to be turned around right away.

Operations such as bulk library conversion are probably best done with lower priority so that they can run at a consistent pace but not hijack availability that might be needed for higher priority jobs.

The priority setting for a single request applies to all jobs generated from the list of files and profiles you supply. So, make sure to use separate requests which encapsulate all of the outputs you want for a given priority.

The default priority is 5. Optional.

job[use_file_cache]

True or False.

Whether or not to have the source file cached in HD Cloud after initial ingest.

The default is to use file caching. NOTE: By setting to false, source file ingest will count against your total bandwidth allowance, which may result in additional charges. Optional.

job[name]

Optional.

job[callback_url]

We'll POST a JSON payload representing your job to this location if given. Optional.

files[]

An array of filenames within the store identified by job[source_id].

encoding_profile_ids[]

An array of integers referring to Encoding Profile IDs.

ERRORS

If your request is invalid, you'll receive an error response. This response will have an *HTTP 403 Forbidden* response code and an errors object in the response body.

XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<errors type="array">
  <error>
    <code type="integer">107</code>
    <message>files are required</message>
  </error>
  <error>
    <code type="integer">105</code>
    <message>encoding_profile_ids are required</message>
  </error>
  <error>
  </error>
</errors>
```

```
<code type="integer">100</code>
<message>job is required</message>
</error>
</errors>
```

JSON:

```
{"errors":
  [{"message": "files are required", "code": 107},
  {"message": "encoding_profile_ids are required", "code": 105},
  {"message": "job is required", "code": 100}]}
```

SUCCESS RESPONSE

XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<jobs type="array">
  <job>
    <id>1</id>
    <current_step>waiting_for_file</current_step>
    <current_progress></current_progress>
    <current_status>waiting for file</current_status>
    <source_filename>beer-drinking-pig.mpg</source_filename>
    <status_url>http://example.com/api/v1/jobs/1</status_url>
  </job>
</jobs>
```

JSON:

```
[{"job":
  {"source_filename": "beer-drinking-pig.mpg",
  "current_step": "waiting_for_file",
  "status_url": "http://example.com/api/v1/jobs/1",
  "id": 1,
  "current_status": "waiting for file",
  "current_progress": nil}}]
```

Show

GET <http://hdcloud.com/api/v1/jobs/:id>

Get a single job. The returned structure will have information about the job including current status, information about associated images(like frame captures), etc.

PARAMETERS

id

An integer referring to a Job ID.

FAILURES

Jobs may fail for a variety of reasons. When this happens, you'll find your jobs marked as failed and two additional elements will be provided: `failure_code` and `failure_message`.

FAILURE CODE	FAILURE MESSAGE
201	Download failed
211	Invalid source file
212	Missing source file
221	Encode stalled
231	Invalid output file
231	Missing output file
241	Delivery failed
500	Generic application error. Unexpected failure

The actual messages you receive may vary. The ones listed here are meant to explain generally what a given code would be used for.

EXAMPLE RESPONSES

XML:

```
<job>
  <id>1</id>
  <current_step>encoding</current_step>
  <current_progress>42</current_progress>
  <current_status>Encoding: Pass 1</current_status>
  <source_filename>beer-drinking-pig.mpg</source_filename>
  <status_url>http://example.com/api/v1/jobs/1</status_url>
</job>
```

JSON:

```
{"job":
  {"source_filename": "beer-drinking-pig.mpg",
    "current_step": "encoding",
    "status_url": "http://example.com/api/v1/jobs/1",
    "id": 1,
    "current_status": "Encoding: Pass 1",
    "current_progress": 42}}
```

Frames

GET <http://hdcloud.com/api/v1/jobs/:id/frames>

Get a list of filenames of all captured frames from a job. Paginated.

Destroy

DELETE <http://hdcloud.com/api/v1/jobs/:id>

Delete a job.

PARAMETERS

id

An integer referring to a Job ID.

Destroy Bulk

DELETE http://hdcloud.com/api/v1/jobs/destroy_bulk

Delete one or more jobs.

PARAMETERS

job_ids[]

An array of Job IDs.

Store Resources

All

GET <http://hdcloud.com/api/v1/stores>

Get all your stores.

Show

GET <http://hdcloud.com/api/v1/stores/:id>

Get a single store.

PARAMETERS

id

An integer referring to a Store ID.

EXAMPLE RESPONSES

XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<store>
  <auth_token nil="true"></auth_token>
  <authorized type="boolean">>false</authorized>
  <cdn>S3</cdn>
  <description>Example Store</description>
  <host nil="true"></host>
```

```
<name>Example Store</name>
<password>cba321</password>
<ref>example-bucket</ref>
<username>123abc</username>
</store>
```

JSON:

```
{"store":
{"name": "Example Store",
"cdn": "S3",
"username": "123abc",
"auth_token": nil,
"description": "Example Store",
"deleted": false,
"host": nil,
"ref": "example-bucket",
"user_id": 1,
"authorized": false,
"password": "cba321"}}
```

Profile Resources

All

GET http://hdcloud.com/api/v1/encoding_profiles

Get all your profiles.

EXAMPLE RESPONSES

XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<encoding_profiles type="array">
  <encoding_profile>
    <audio_sample_rate type="integer">44100</audio_sample_rate>
    <audiobitrate type="integer">96</audiobitrate>
    <bitrate>500400300</bitrate>
    <bitrate_tolerance type="integer" nil="true"></bitrate_tolerance>
    <brightcove type="boolean">>false</brightcove>
    <brightcove_publisher_id nil="true"></brightcove_publisher_id>
    <created_at type="datetime" nil="true"></created_at>
    <description>Example Profile</description>
    <framerate type="float">30.0</framerate>
    <keyframe_interval type="integer" nil="true"></keyframe_interval>
    <letterbox type="boolean">>true</letterbox>
    <max_file_size type="float" nil="true"></max_file_size>
    <name>Example Profile</name>
    <output_ext nil="true"></output_ext>
    <resolutions>400x300</resolutions>
    <thumbnail_interval type="float" nil="true"></thumbnail_interval>
```

```
<thumbnail_offset type="float" nil="true"></thumbnail_offset>
<time_duration type="integer" nil="true"></time_duration>
<time_offset type="integer" nil="true"></time_offset>
<updated_at type="datetime" nil="true"></updated_at>
</encoding_profile>
</encoding_profiles>
```

JSON:

```
{"encoding_profiles":
  [{"name": "Example Profile",
    "resolutions": [["400x300"]],
    "audiobitrate": 96,
    "brightcove_publisher_id": nil,
    "updated_at": nil,
    "framerate": 30.0,
    "brightcove": false,
    "output_ext": nil,
    "keyframe_interval": nil,
    "audio_sample_rate": 44100,
    "bitrate_tolerance": nil,
    "letterbox": true,
    "time_offset": nil,
    "bitrate": [500, 400, 300],
    "description": "Example Profile",
    "thumbnail_interval": nil,
    "thumbnail_offset": nil,
    "max_file_size": nil,
    "created_at": nil,
    "time_duration": nil}]}
```

Show

GET http://hdcloud.com/api/v1/encoding_profiles/:id

Get a single profile.

PARAMETERS

id

An integer referring to a Profile ID.

EXAMPLE RESPONSES

XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<encoding_profile>
  <audio_sample_rate type="integer">44100</audio_sample_rate>
  <audiobitrate type="integer">96</audiobitrate>
  <bitrate>500400300</bitrate>
  <bitrate_tolerance type="integer" nil="true"></bitrate_tolerance>
  <brightcove type="boolean">>false</brightcove>
  <brightcove_publisher_id nil="true"></brightcove_publisher_id>
```

```
<created_at type="datetime" nil="true"></created_at>
<description>Example Profile</description>
<framerate type="float">30.0</framerate>
<keyframe_interval type="integer" nil="true"></keyframe_interval>
<letterbox type="boolean">true</letterbox>
<max_file_size type="float" nil="true"></max_file_size>
<name>Example Profile</name>
<output_ext nil="true"></output_ext>
<resolutions>400x300</resolutions>
<thumbnail_interval type="float" nil="true"></thumbnail_interval>
<thumbnail_offset type="float" nil="true"></thumbnail_offset>
<time_duration type="integer" nil="true"></time_duration>
<time_offset type="integer" nil="true"></time_offset>
<updated_at type="datetime" nil="true"></updated_at>
</encoding_profile>
```

JSON:

```
{"encoding_profile":
{"name": "Example Profile",
"resolutions": [["400x300"]],
"audiobitrate": 96,
"brightcove_publisher_id": nil,
"updated_at": nil,
"framerate": 30.0,
"brightcove": false,
"output_ext": nil,
"keyframe_interval": nil,
"audio_sample_rate": 44100,
"bitrate_tolerance": nil,
"letterbox": true,
"time_offset": nil,
"bitrate": [500, 400, 300],
"description": "Example Profile",
"thumbnail_interval": nil,
"thumbnail_offset": nil,
"max_file_size": nil,
"created_at": nil,
"time_duration": nil}}
```