



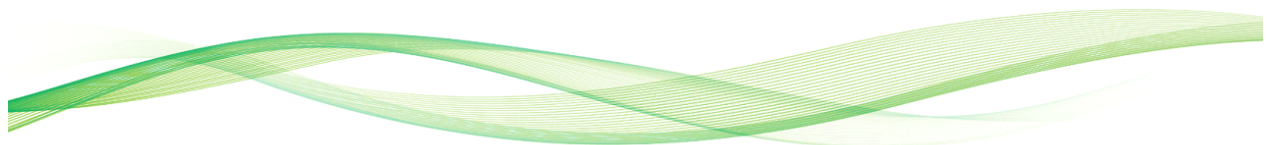
# Vantrix Corporation

## VTA QuickStart

---

**Version:** 56

**Date:** June 17, 2016



This material and information (“Information”) constitutes a trade secret of Vantrix Corporation (“Vantrix”) and is strictly confidential. You agree to keep this information confidential and to take all necessary measures to maintain its secrecy. Without limiting the foregoing, Vantrix Corporation considers its confidential information, including, but not limited to any source code and technical information, to be an unpublished proprietary trade secret. If an authorized publication occurs, the following notice shall be affixed to it: Copyright © 2016 Vantrix Corporation. All Rights Reserved.

No part of this material may be reproduced, including, but not limited to, photocopying, electronic, or mechanical recording, nor stored in a retrieval system, or otherwise transmitted, in any form or by any means, without the prior written permission of Vantrix Corporation.

Vantrix Corporation assumes no responsibility for any errors or omissions. This Information is subject to continuous updates and improvements. All warranties implied or expressed, including but not limited to implied warranties of merchantability, fitness for purpose, condition of title, and non-infringement, are specifically excluded. In no event shall Vantrix Corporation and its suppliers be liable for any special, indirect, or consequential damages or any damages whatsoever arising out of or in connection with the use of this information. The foregoing disclaimer shall apply to the maximum extent permitted by applicable law, even if a particular remedy fails its essential purpose.

Vantrix is either a registered trademark or a trademark of Vantrix Corporation in Canada and/or other countries. Any unauthorized use is strictly prohibited.

All other names and trademarks are the property of their respective owners.

Any reference in the text of this document to VoiceAge Networks shall be construed as a reference to Vantrix.

Copyright © 2016 Vantrix Corporation

Vantrix Corporation  
1425 Rene-Levesque West, Suite 1200  
Montreal, Quebec H3G 1T7 CANADA  
Tel: +1 (514) 866-1717  
Fax: +1 (514) 866-6868  
<http://www.vantrix.com/>

# Contents

1	Basic concepts .....	5
2	Getting started in Bluemix .....	7
2.1	From the Bluemix GUI .....	7
3	Trying it .....	11
3.1	Step 1. Upload a media .....	11
3.2	Step 2. Define the transcoding criteria .....	12
3.3	Step 3. Creating a new preset .....	13
3.4	Step 3. Defining the job object .....	15
3.5	Step 4. Submit the job object to VTA service .....	19
3.6	Step 5. Wait for completion .....	20
3.7	Step 6. Download the result .....	22

# Introduction



This is a step-by-step set of instructions to guide you through your first series of interactions with the Vantrix Transcoding API (VTA) and bring you to some level of comfort. This guide does not use any specific language to execute the commands but does provide the equivalent *curl* commands that can be used from the shell prompt (e.g., in a Mac or Linux environment) to perform the REST-based service calls.

# 1 Basic concepts

Before we start, let's review a few basic concepts.

## Service endpoints

The VTA API provides two service endpoints:

1. `"/vta/v1/presets"` for preset management
2. `"/vta/v1/jobs/transcode"` for transcoding job control

It also includes an online API accessible from the UI.

## Source and result storage

Before transcoding can be done using the API (or UI), a file must be uploaded into the cloud service source library. Uploading can be done using the UI or the secure FTP service (more on this in the step-by-step instructions below). The transcoding result will also be stored in the cloud, to be downloaded later. The user is responsible for managing these files, that is, for deleting the sources or results when they aren't needed anymore.

## Job model

The API uses an asynchronous job model. The user (i.e., application) defines the job criteria including the input, transcoding options and output location and then uses an HTTP POST (REST) to submit the transcoding job to the cloud service. From this point, the application can poll the system for status (get the job description) or wait for a notification (HTTP callback).

A job is considered completed when it has reached a completion state. There are three completion states: *completed*, *stopped* and *error*.

## Transcoding criteria (presets)

The job transcoding criteria (formats and codecs) can be fully qualified in the job object, or the job can refer to an existing preset by name. When a preset is used, all its properties will be applied on the job definition but only if the property is not already defined in the job. This mechanism allows the user to override some preset properties and provides additional flexibility when needed.

## Presets

A preset is a collection of transcoding properties that are persisted and given a name for reuse. Presets come in two flavors: *system* and *user*. A *system* preset is read-only and built into the system. A *user* preset is any preset added by a user. The differences stop there. Both preset types can be referred to and used exactly the same way in a transcoding job.

### Access Key

An access key is obtained during the service subscription phase and required for interacting with the VTA API. The key is private to the user and must not be shared. It authorizes a user and a request to interact with the service. The key is bundled with the request using the "x-vta-key" HTTP header. If you don't have your key, you can get it from the VTA UI "Account" page. Open the VTA UI using the Bluemix service dashboard option/button and select the "Account" option from the navigation panel. At the bottom of the page are the FTP, storage and API access points and credentials.

### Access Points

The service provides three different access points. The URLs and credentials are all displayed at service creation and also on the "Account" page of the service UI accessible from the service dashboard link in Bluemix.

1. Secure FTP: provides direct secure FTP access to the storage
2. Storage: provides direct Swift-based access to the storage
3. VTA: provides a URL and access key for API usage

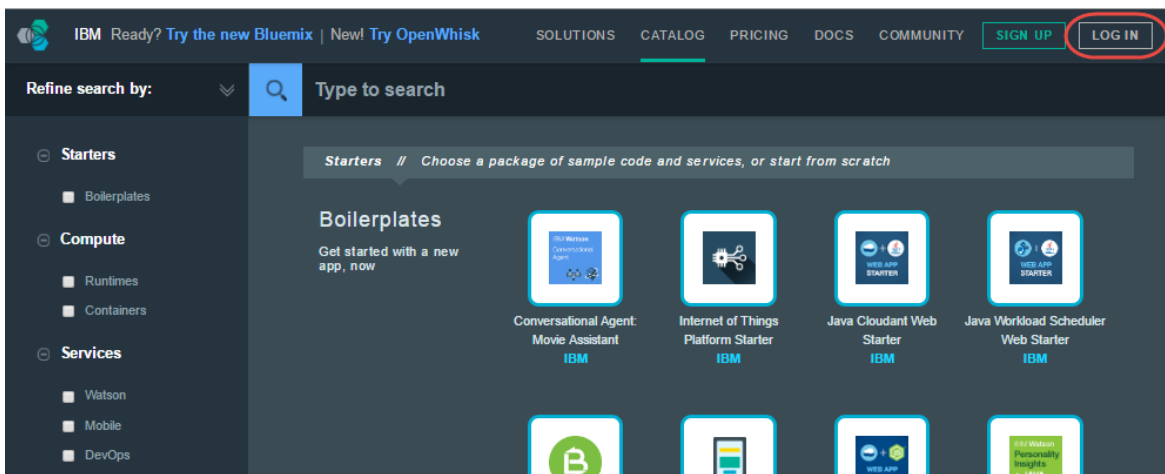
Now let's proceed with concrete samples.

## 2 Getting started in Bluemix

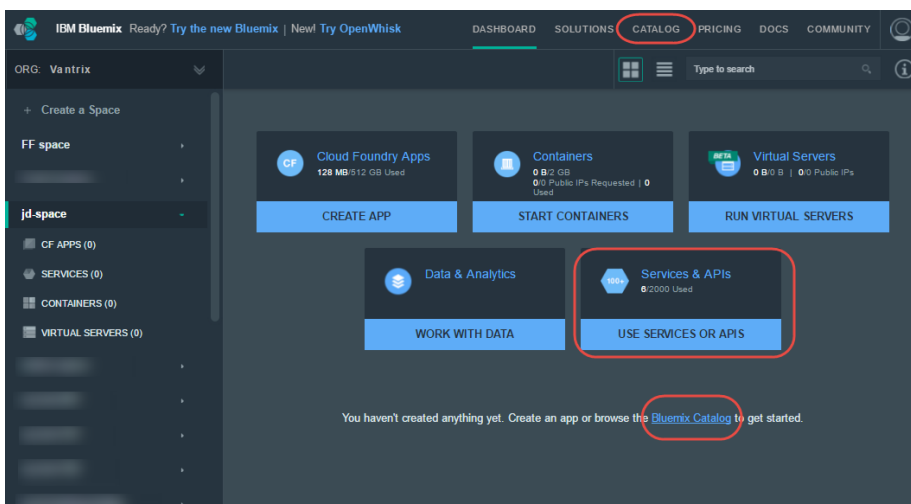
This section explains how to log into Bluemix and create a Vantrix service.

### 2.1 From the Bluemix GUI

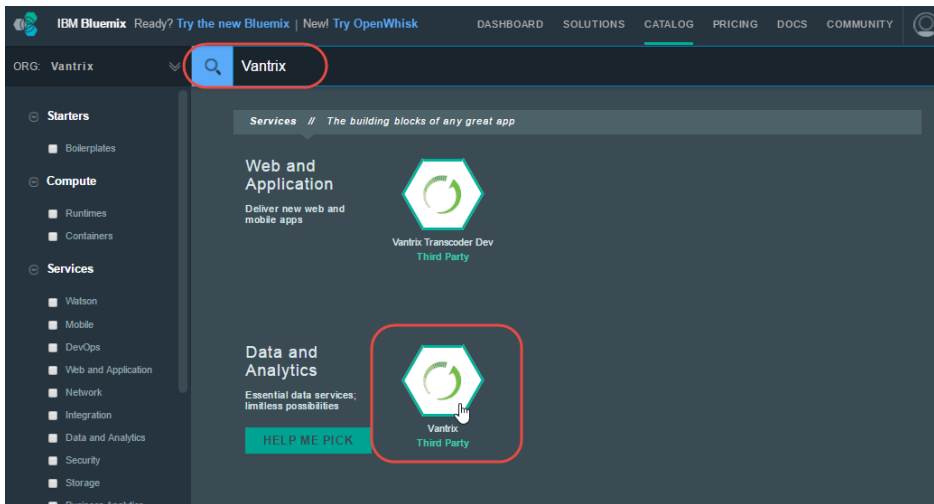
1. In a browser, enter <https://console.ng.bluemix.net/catalog>
2. In the top right corner of the window, click **Log in** and enter your Bluemix credentials. (If you don't already have a Bluemix account, click Sign up to create an account.)




3. Go to the Bluemix catalog: click **Catalog** in the menu bar or in the message below the tiles, or click the Services and APIs tile.



4. In the catalog **Search** box, enter Vantrix, and then click the Vantrix tile.



5. In the service description / catalog entry, click **CREATE**.



**Vantrix**  
Third Party

PUBLISH DATE  
06/07/2016

AUTHOR  
Vantrix


TYPE  
Service

[VIEW DOCS](#)

The Vantrix Transcoding API gives you the tools to transcode, package and deliver live media and videos so they reach the widest audience and provide the best end-user experience.

Pick a plan Monthly prices shown are for country or region: [Canada](#)

Plan	Features	Price
✓ Vantrix Cloud FREE	File transcoding 10GB storage, max 10 files 1 concurrent job, output limited to 30 seconds Full GUI and API access	Free

 Video Transcoding

[TERMS](#)

Note: Subscription discounts don't apply to third-party services.

Add Service

Space:

App:

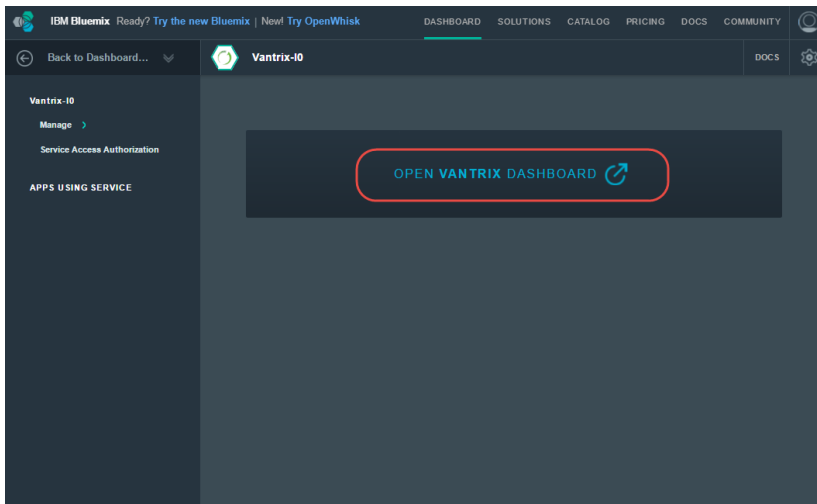
Service name:

Selected Plan:

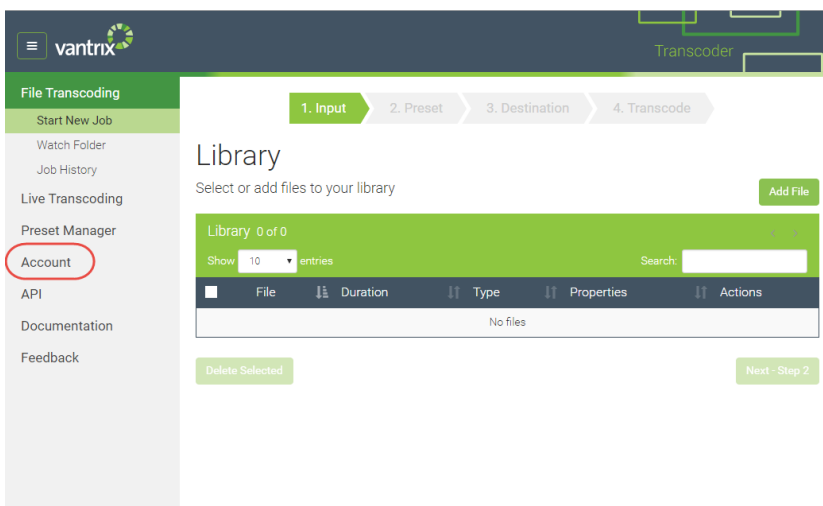
[CREATE](#)



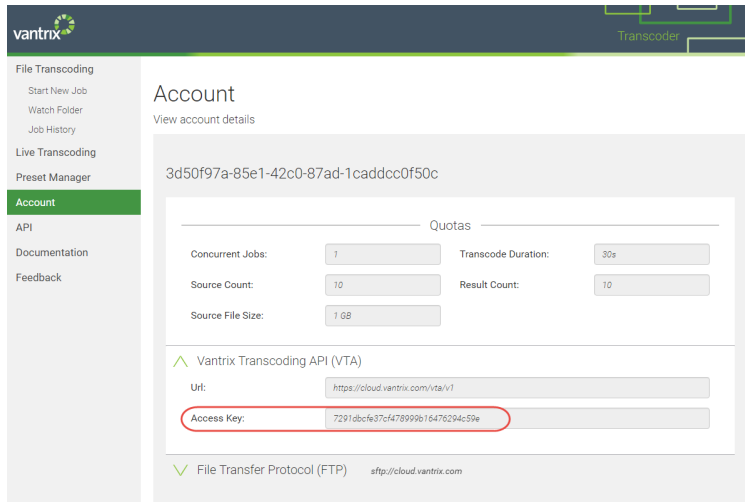
6. Click **Open Vantrix Dashboard**.



The Transcoder Cloud UI opens.



7. To obtain your access key, click **Account** and on the Account page, expand the **Vantrix Transcoding API (VTA)** section.



The screenshot shows the Vantrix web interface. On the left is a navigation menu with options like 'File Transcoding', 'Live Transcoding', and 'Account'. The main content area is titled 'Account' and shows account details. A 'Quotas' section lists settings for concurrent jobs, source count, and file size. Below that, the 'Vantrix Transcoding API (VTA)' section is expanded, showing the 'Url' and 'Access Key'. The 'Access Key' value is highlighted with a red circle. At the bottom, there is a section for 'File Transfer Protocol (FTP)'.

Quotas	
Concurrent Jobs:	1
Transcode Duration:	30s
Source Count:	10
Result Count:	10
Source File Size:	1 GB

Vantrix Transcoding API (VTA)

Url: <https://cloud.vantrix.com/vta/v1>

Access Key: 7291abcfe37cf478999b16476294c59e

File Transfer Protocol (FTP) [#ftp://cloud.vantrix.com](ftp://cloud.vantrix.com)

8. Use your access key with every call to the VTA. This page also lists your secure FTP URL and login credentials.

## 3 Trying it

### 3.1 Step 1. Upload a media

First, you need to upload your media into the cloud. Transcoding is always done on a file located in the cloud, and the result is also put in the cloud.

There are two ways to upload a media:

1. Upload your media using the UI source Library page, or
2. Upload the file using the secure FTP service. This way is more convenient when building an application and controlling the upload process from your app.

In both cases, the file will be added to your source library folder in the cloud. This is a single folder holding all the files that are candidates for transcoding. It is up to the user to add and delete files in this folder. For a file to be transcoded from the UI or API, it must be in the cloud in this folder.

#### Uploading using SFTP

1. First of all, get the credentials from the UI Account page in the corresponding Access Point. You will have a host, username and password.
2. Open your preferred SFTP client, or use your shell client if you have one.
3. Connect to the SFTP host and authenticate. You will be brought to your space.
4. You will see two folders, *sources* for the sources library and *results* for the transcoding results.
5. Upload any file into *sources*.

**Try it - connect to storage using sftp**

```
; Assuming username is 12345. Connect using the shell sftp client
sftp 12345@cloud.vantrix.com
; Then enter the password
12345@cloud.vantrix.com's password:
; You will see the following prompt
Connected to cloud.vantrix.com.
sftp>
; Do a remote file listing
sftp> ls -la
drwx-----    1 65535    65535                0 Apr 08 17:14  .
drwx-----    1 65535    65535                0 Apr 08 17:14  ..
drwx-----    1 65535    65535            0 Apr 08 17:14  results
drwx-----    1 65535    65535            0 Apr 08 17:14  sources
; Move into sources
sftp> cd sources
; Upload your media file
sftp> put bigbuckbunny-hd.mp4
```

## 3.2 Step 2. Define the transcoding criteria

The next step is to set the transcoding criteria to be used in the transcoding job object to qualify the output. If you are executing a common transcoding task, you may already have a preset available and qualified for the transcoding you want to apply on your media. In order to search and find a matching preset, you can view the presets list from the UI or you can request a list using the API. You can also filter the list on any criteria (e.g., list all presets having HD resolution).

### Getting the preset list from the API

View all presets by submitting a service GET call on the *presets* resources. Any filtering options would be added as parameters (refer to the API specification). Don't forget to add the *x-vta-key* header with the context key.

*curl* command:

**Try it - get all presets**

```
curl -X GET -H "x-vta-key: mykey" "https://cloud.vantrix.com/vta/v1/presets"
```

*Note: for every service call, make sure you replace "mykey" with your own context key.*

You will get back a "200 OK" and a list of preset objects so you can determine if one is a good fit for the task.

### 3.3 Step 3. Creating a new preset

If no preset is available for the specific task you want to execute, you can choose to add a new one to the collection. When adding a preset, we assume it is likely to be reused later.

Otherwise, we could choose to specify all the properties in the job definition. Presets are also very convenient in an automated process, where you can change a preset property and not have to rebuild/redeploy the application. Simply update the preset and submit a new job.

#### Adding an SD preset

Suppose that no preset is applicable to your task and you would like to persist the set of transcoding options as a new preset for reuse. You add a preset by submitting a REST HTTP POST with the preset object to the VTA service. See the example below. The preset defines the output criteria only and maps almost one-to-one to the job output object. We'll cover that below.

Example:

```
POST http://<cloud-host>:12200/vta/v1/presets
Content-type: application/json
x-vta-key: mykey
Body:
{
  "name": "SD",
  "format": {
    "type": "hls"
  },
  "tracks": [
    {
      "audio": [
        {
          "codecType": "aac",
          "bitrate": 96000,
          "samplingRate": 48000,
          "channelMode": "stereo",
          "sbr": false,
          "passthrough": false
        }
      ],
      "video": {
        "codecType": "h264",
        "qualityLevel": 5,
        "bitrate": 3500000,
        "width": 960,
        "height": 540,
        "kfs": 2000,
        "framerate": "30000 / 1001"
      }
    }
  ]
}

=> HTTP 201 Created
```

And the *curl* command:

**Try it - create SD preset**

```
curl -X POST -H "Content-Type: application/json" -H "x-vta-key:
mykey" -d '{
  "name": "SD",
  "format": {
    "type": "hls"
  },
  "tracks": [
    {
      "video": {
        "codecType": "h264",
        "qualityLevel": 5,
        "bitrate": 3500000,
        "width": 960,
        "height": 540,
        "kfs": 2000,
        "framerate": "30000 / 1001"
      },
      "audio": [
        {
          "codecType": "aac",
          "bitrate": 96000,
          "samplingRate": 48000,
          "channelMode": "stereo",
          "sbr": false,
          "passthrough": false
        }
      ]
    }
  ]
}' "https://cloud.vantrix.com/vta/v1/presets"
```

## 3.4 Step 3. Defining the job object

The job object has three main components:

- The input location and definition
- The output location
- The transcoding criteria

You can also provide an HTTP callback URL to subscribe to a notification when the job completes. Please refer to the API online documentation or API Specification document for more details.

### The input definition

The *input* mainly defines the URL to the file resource located in the cloud source library, for example, `file:///myvideo.mp4`. You may want to provide additional details and override decoding instructions, but in most cases the URL is sufficient.

The JSON input main properties are:

```
"input": {
  "url": String,
  "type": [vod, live, loop],
  "audio": { ... },
  "video": { ... }
}
```

The *url* is mandatory and needs to refer to a file in the source library. The *audio* and *video* are optional and will carry decoding instructions only if the defaults are not sufficient or an override of the defaults is needed. The *type* will be "vod" for file transcoding and "live" for streams. A special "loop" is also supported to simulate a live feed from a file.

### The output definition

The output location tells where to create the transcoding result. It contains a URL and format information (hls, mp4, etc.).

The transcoding criteria define how to transcode the file in terms of codecs, resolution, bitrate and so on. If you found a matching preset, you can simply specify the preset name in the job output definition.

The most important JSON properties of the output object are:

```
"outputs": [
  {
    "presetName": String,
    "format": { ... },
    "target": { ... },
    "tracks": [
      {
        "audio": [
          { ... }
        ],
        "texts": [
```



```

        { ... }
      ],
      "video": { ... }
    }
  ],
}
]

```

The *format* is mandatory and defines the file container (mp4, hls, mpegts) and related properties. Minimally, we need a type in there. The *presetName* is optional and allows the use of an existing preset if one is fit for the job. The *target* is also mandatory and must include a URL to the output file or stream. The *tracks* property is an array of *audio* and *video* tracks. It also supports a *texts* track in case we are dealing with subtitles. The *tracks* property is optional if a *presetName* is provided and defines all the track objects.

Note:

The VTA API supports multiple outputs. We won't cover this here but it can be used to produce either multiple unrelated single-file contents or multiple HLS variants of a content.

Example: a simple job with a preset

Since we have created an "SD" preset, we'll use it.

```

{
  "input": {
    "url": "file:///bigbuckbunny-hd.mp4"
  },
  "outputs": [
    {
      "presetName": "SD",
      "target": {
        "url": "file:///bigbuckbunny/bigbuckbunny.m3u8"
      }
    }
  ]
}

```

In the sample above, we have a very simple structure for transcoding a movie from HD to SD and changing the media container from mp4 to HLS. The SD preset contains all the track and format definitions. If we read the job after it is submitted, we will see that all the details of the preset were copied into the job definition at creation time. Basically, the preset is used only to quickly fill the *format* and *tracks* of a given job.

Example: a fully qualified job definition

In the following sample, the user didn't find a suitable preset nor does he want to add one for this job. He must therefore fill the *format* and *tracks* properties.

```
{
  "input": {
    "url": "file:///bigbuckbunny-hd.mp4"
  },
  "outputs": [
    {
      "target": {
        "url": "file:///bigbuckbunny/bigbuckbunny.m3u8"
      },
      "format": {
        "type": "hls"
      },
      "tracks": [
        {
          "audio": [
            {
              "codecType": "aac",
              "bitrate": 96000,
              "samplingRate": 48000,
              "channelMode": "stereo",
              "sbr": false,
              "passthrough": false
            }
          ],
          "video": {
            "codecType": "h264",
            "qualityLevel": 5,
            "bitrate": 3500000,
            "width": 960,
            "height": 540,
            "kfs": 2000,
            "framerate": "30000 / 1001"
          }
        }
      ]
    }
  ]
}
```

## 3.5 Step 4. Submit the job object to VTA service

Once the job object is defined, the client needs to submit it to the VTA service. This is done using a REST HTTP POST to the VTA API service endpoint. Once again the *x-vta-key* is required, as well as a header indicating the content-type being posted is of type *application/json*.

### Getting the object back on create

Unless specifically requested, the VTA service won't return the full job JSON object stored in the system. If a client needs to get this back, it needs to add the "*returnobj=true*" parameter.

### Submitting the job

Following the standard HTTP-based REST rules, the VTA service will return a "201 Created" status if all went well. If anything is wrong with the validation of the request, a "400 Client error" will be returned with violation details. If you are using a limited plan, you may get "429 Limit exceeded" if you are using more than the allocated resources.

Example:

```
POST http://<cloud-host>:12200/vta/v1/jobs
Content-type: application/json
x-vta-key: mykey
Body:
{
  "input": {
    "url": "file:///bigbuckbunny-hd.mp4"
  },
  "outputs": [
    {
      "target": {
        "url": "file:///bigbuckbunny/bigbuckbunny.m3u8"
      },
      "presetName" : "SD"
    }
  ]
}

=> HTTP 201 Created
```

*curl* command:

**Try it - create a transcoding job**

```
curl -X POST -H "Content-Type: application/json" -H "x-vta-key:
mykey" -d '{
  "input": {
    "url": "file:///bigbuckbunny-hd.mp4"
  },
  "outputs": [
    {
      "presetName" : "SD",
      "target": {
        "url": "file:///bigbuckbunny/bigbuckbunny.m3u8"
      }
    }
  ]
}' "https://cloud.vantrix.com/vta/v1/jobs/transcode"
```

## 3.6 Step 5. Wait for completion

The VTA *jobs* service follows the "fire and forget" model. But you can poll the job using its ID at any moment to monitor its current status. The job object will be returned with its status.

Moreover, you could include a notification callback URL in the job definition at creation time and simply wait for the notification. Both options are valid.

If the job is found, a "200 OK" is returned with the job object. Otherwise, a "404 Not Found" is returned.

### Getting the job status

```
GET http://cloud.vantrix.com/vta/v1/jobs/5704c26a37cc60f5c9818a66
==> HTTP 200 Ok
{
  "id": "5704c26a37cc60f5c9818a66",
  "type": "transcode",
  "priority": 5,
  "input": {
    "url": "file:///bigbuckbunny-hd.mp4"
    "type": "vod"
  },
  "outputs": [
    {
      "target": {
        "url": "file:///bigbuckbunny/bigbuckbunny.m3u8",
        "method": "post"
      },
      "presetName": "SD",
```

```
    "tracks": [
      {
        "audio": [
          {
            "codecType": "aac",
            "bitrate": 96000,
            "samplingRate": 48000,
            "channelMode": "stereo",
            "sbr": false,
            "passthrough": false
          }
        ],
        "video": {
          "codecType": "h264",
          "qualityLevel": 5,
          "bitrate": 3500000,
          "width": 960,
          "height": 540,
          "kfs": 2000,
          "framerate": "30000 / 1001",
          "scalingMethod": "preserve",
          "afdEnable": false,
          "advancedParams": "profile=baseline",
          "gpuAcceleration": false,
          "passthrough": false
        }
      }
    ],
    "format": {
      "type": "hls",
      "programId": 1,
      "offset": 20,
      "segmentDuration": 10000,
      "segmentDurationPrecision": 0,
      "segmentBufferSize": 10
    }
  ],
  "status": "submitted",
  "creationTime": "2016-04-06T04:01:46.627-0400"
}
```

You'll notice a few more fields such as *status*, *creationTime*, *priority*, etc. Some are defaults added by VTA, some are additional status information such as characterization and processing info and some are time stamps.

After posting a new job, you will get back a location header indicating how to reach this job object for status. If you don't get it, you can still get the full list or try filtering to reduce the set (refer to the API specification). For example, the following *curl* command will get all jobs having a preset with the name "HD".

**Try it - get all job objects**

```
curl -X GET -H "Content-Type: application/json" -H "x-vta-key: mykey" "https://cloud.vantrix.com/vta/v1/jobs/transcode?outputs.presetName=HD"
```

## 3.7 Step 6. Download the result

When the job status is *completed*, the next step is to download the resulting transcoding output. According to the format, you have to download a single file or a whole folder of playlists and fragments.

You can download the content from the UI (which will zip a multipart content) or use the secure FTP access point.

Refer to the *step 1-upload using sftp* section, but select the *results* folder. This folder will contain one folder with the content per transcoding job.