

uStudio Destination API

Overview

The uStudio Destination API enables an application to act on behalf of a uStudio user to create new destinations, publish videos to those destinations, and unpublish videos from those destinations. The intent of the system is to provide a simple, parallel process for managing videos across a wide variety of endpoints, including custom integrations.

The uStudio API Overview should be read first, in order to understand basic authentication methods, resource types, etc. Please contact support if you do not have access to that document.

Authentication

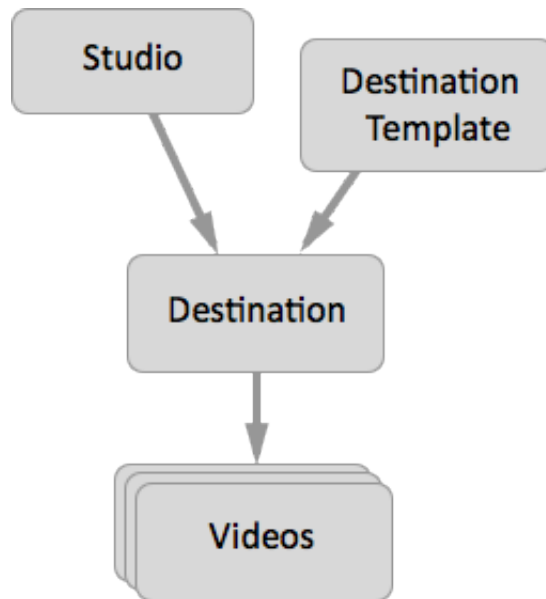
Operations on the Destination API should be authorized through a user token. The method for obtaining a user token is described previously. Additionally, the application being used to obtain the user token must have access to the destination API. For any restricted access issues, please contact support to ensure your application has the appropriate permissions.

Since each action is taken on behalf of a user, the operation is limited to the permissions of that user. For instance, if a specific destination is available only to the owner of a studio, a token for a member of that studio will not allow access to operations on that resource.

Overview of Destinations

In uStudio, Destinations are attached to a Studio. Each Destination is produced from a Destination Template, and inherits the characteristics of that template. For instance, the Embeddable Player Destination

Template ensures every video published to that destination generates all player transcodes, stores them properly on the configured CDN, and exposes an embed code.



Destination Relationships

Destination Templates cannot be created – they are associated with a Studio based on the Studio’s plan, the Application permissions, and other factors. Custom Destination Templates can be built for Studios; for more information on services, please contact support.

Each Video published to a Destination retains its metadata. Based on the Destination Template, that metadata is used when submitting to external APIs, building HTML pages, etc. Additional data is exposed on the video as necessary in the form of a Destination Video, which is covered in another section.

Determining the Destinations API Endpoint

Every Studio has a Destinations endpoint URL. It usually looks like the following:

```
https://app.ustudio.com/api/v2/studio/STUDIOUID/destinations
```

However, it is almost always preferable, when possible, to use the URLs provided by the API. This prevents changes in uStudio's infrastructure from affecting your software application. When you retrieve a studio resource (described previously), the destinations URL is in the response body:

```
{
  "collections_url": "/api/v2/studios/Ok82U2krWVzp/collections",
  "created": "2011-12-20 17:11:26",
  "destination_templates_url": "/api/v2/studios/Ok82U2krWVzp/destination_templates",
  "destinations_url": "/api/v2/studios/Ok82U2krWVzp/destinations",
  "name": "uStudio Demo",
  "studio_url": "/api/v2/studios/Ok82U2krWVzp",
  "uid": "Ok82U2krWVzp",
  "videos_url": "/api/v2/studios/Ok82U2krWVzp/videos"
}
```

The full destination API endpoint would be:

```
https://app.ustudio.com/api/v2/studios/Ok82U2krWVzp/destinations
```

As mentioned before, always check to make sure the host is not already part of the URL, as we are transitioning from relative URLs to absolutes.

Creating a Destination

To create a Destination, you'll need a Destination Template UID. These do not change, so you'll only need to get it once for each Destination Template type you want to create. To see a list of Destination Templates (and their UIDs) simply perform a `GET` request to the `destination_templates_url` provided in the Studio response. (Make sure to provide the token.) Below is an example response:

```
{
  "destination_templates": [
    {
      "action": "authorize",
      "destination_template_url": "/api/v2/studios/0k82U2krWVzp/destination_templates/LYV884n5txM7",
      "name": "Video Site",
      "tag": "pages",
      "uid": "LYV884n5txM7"
    },
    {
      "action": "authorize",
      "destination_template_url": "/api/v2/studios/0k82U2krWVzp/destination_templates/LqxzBLtIZQtg",
      "name": "Embeddable Player",
      "tag": "player",
      "uid": "LqxzBLtIZQtg"
    }
  ]
}
```

The Destination Template UID for the Embeddable Player is `LqxzBLtIZQtg`.

The process of creating a Destination is tied to the requirements of a Destination Template. Some Destination Templates (like the Embeddable Player) are quite simple and require no input from the end user. Other Destination Templates such as YouTube or Facebook require authorization prompts and other user-facing steps.

All Destinations are created by making a `POST` request to the Destinations endpoint discussed above. The content of the request is based on the Destination Template type. Below is the example request body of a new Embeddable Player:

```
METHOD: POST
URL: https://app.ustudio.com/api/v2/studios/0k82U2krWVzp/destination_templates/
LqxzBLtIZQtg
REQUEST BODY:
{
  "destination_template_uid": "LqxzBLtIZQtg",
  "enable_beta_player": true,
  "enable_watermark": false,
  "name": "New Player",
  "publish_webhook": "",
  "unpublish_webhook": "",
  "update_webhook": ""
}
```

All new Destinations require a `name` field to identify the Destination to other users of the Studio. Additionally, you must provide a `destination_template_uid` – in this case, it is the same UID we retrieved previously for a Player. The other options depend on the destination we are creating. In this instance, we are turning off the watermark for this player, enabling the new HTML5 beta player, and disabling the Publish API hooks.

A successful response code is either a `201` or `202`. A `201` indicates that the Destination was created successfully and requires no additional authorization from the user. (`202` response codes are covered in the next section.) Examples of Destinations returning `201` codes are Embeddable Players, Video Sites, iTunes Feeds, etc.

A `201` response will contain information necessary for additional operations. Here is the response to the new player request:

```
{
  "destination_url": "/api/v2/studios/0k82U2krWVzp/destinations/DSDqOPFrVbbi"
,
  "enable_beta_player": true,
  "enable_watermark": false,
  "files": {
    "upload_url": "https://upload01-app.ustudio.com/api/v2/studios/0k82U2krWVzp/destinations/DSDqOPFrVbbi/files",
    "watermark": null
  },
  "name": "New Player",
  "publish_webhook": "",
  "studio_uid": "0k82U2krWVzp",
  "studio_url": "http://app.ustudio.com/api/v2/studios/0k82U2krWVzp",
  "tag": "player",
  "uid": "DSDqOPFrVbbi",
  "unpublish_webhook": "",
  "update_webhook": "",
  "videos_url": "/api/v2/studios/0k82U2krWVzp/destinations/DSDqOPFrVbbi/videos"
}
```

The first attribute an application must track is the `destination_url`. This is the base access point for updating and deleting the Destination, in addition to retrieving the Destination representation again. A `GET` to this URL will return a similar response.

The next attribute of interest is `videos_url` – this location identifies the location for publishing and retrieving videos published to this Destination. This is used in the Publishing section below.

Authorizing a Destination

Destinations that require external API validation (YouTube, Twitter, etc.) will return a response code of `202`. The response body will be much simpler, but the Destination has not been created yet:

```
{
  "job_url": "/api/v2/studios/0k82U2krWVzp/jobs/JIXeSpRpGIGO",
  "redirect_user_url": "/api/v2/studios/0k82U2krWVzp/destination_templates/L
9QifKezcAYU/authorization/redirect?job_uid=JIXeSpRpGIGO"
}
```

The `job_url` will need to be tracked. A `GET` request to the `job_url` will result in information about the authorization job's state:

```
{
  "failed": false,
  "finished": false,
  "status": "waiting",
  "uid": "JX3CvXQPauzj"
}
```

If `redirect_user_url` is present, that indicates that a user (via a browser) must be directed to that URL. After authorizing uStudio, the authorization check may continue.

A successful authorization will update the job response:

```

{
  "failed": false,
  "finished": true,
  "result": {
    "destination_url": "/api/v2/studios/Ok82U2krWVzp/destinations/DSDqOPFrVbbi",
    "enable_beta_player": true,
    "enable_watermark": false,
    "files": {
      "upload_url": "https://upload01-app.ustudio.com/api/v2/studios/Ok82U2krWVzp/destinations/DSDqOPFrVbbi/files",
      "watermark": null
    },
    "name": "New Player",
    "publish_webhook": "",
    "studio_uid": "Ok82U2krWVzp",
    "studio_url": "http://app.ustudio.com/api/v2/studios/Ok82U2krWVzp",
    "tag": "player",
    "uid": "DSDqOPFrVbbi",
    "unpublish_webhook": "",
    "update_webhook": "",
    "videos_url": "/api/v2/studios/Ok82U2krWVzp/destinations/DSDqOPFrVbbi/videos"
  },
  "status": "finished",
  "uid": "JX3CvXQPauzj"
}

```

Based on response times from the external destination and the size of the Job queue, it may take a small amount of time to authenticate the destination. The `job_url` will need to be polled, in no less than five second increments, until it has succeeded or failed.

A failed authorization will not create a Destination, and the Destination creation process must be restarted.

Retrieving Destinations

A list of the available destinations can be produced simply by performing a `GET` request on the appropriate Destination URL. A simplified representation of every destination associated with the Studio will be returned:


```
{
  "destinations": [
    {
      "destination_url": "/api/v2/studios/Ok82U2krWVzp/destinations/DSDqOPFrVbbi",
      "enable_beta_player": false,
      "enable_watermark": false,
      "name": "New Player (Updated)",
      "publish_webhook": "",
      "studio_uid": "Ok82U2krWVzp",
      "uid": "DSDqOPFrVbbi",
      "unpublish_webhook": "",
      "update_webhook": ""
    }
  ]
}
```

Updating a Destination

Updating a Destination is straightforward: simply make a `PUT` request (to the appropriate Destination URL) containing the attributes to change. This request should be formatted in `JSON`. The following is an example that changes a few settings on the Player we just created:

```
METHOD: PUT
URL: https://app.ustudio.com/api/v2/studios/Ok82U2krWVzp/destinations/DSDqOPFrVbbi
REQUEST BODY:
{
  "enable_beta_player": false,
  "name": "New Player (Updated)"
}
```

Notice you only need to provide the attributes you want to change, not the entire Destination representation. Not all attributes are changeable – changing the UID or any URLs, for instance, are not allowed.

A successful request will result in a `200` response code, with a JSON representation of the Destination as the response body.

Managing a Destination's Files

The Player Destination we created allows a watermark to be uploaded. In the Destination's representation, there is a `files` section which identifies an `upload_url` as well as information about the previously-uploaded files on the Destination.

Some Destination Files require settings on the Destination. For instance, in order for the watermark to be visible on a Player, a Destination must have `enable_watermark` set to `true`.

Uploading file(s) to a Destination simply involves a `multipart/form-data` `POST` request to the `upload_url` identified previously. In the example above, this URL is:

```
https://upload01-app.ustudio.com/api/v2/studios/Ok82U2krWVzp/destinations/DSDqOPFrVbbi/files
```

The details of making a multipart POST are beyond the scope of this document, however make sure that each multipart name corresponds to the proper Destination File you are changing. In this case, the name would be `watermark`, and the file type must be a `JPEG`.

After performing the upload, future `GET` requests on the Destination URL will return information about the file:

```
{
  "destination_url": "/api/v2/studios/Ok82U2krWVzp/destinations/DSDqOPFrVbbi"
,
  "enable_beta_player": false,
  "enable_watermark": false,
  "files": {
    "upload_url": "https://upload02-app.ustudio.com/api/v2/studios/Ok82U2krWVzp/destinations/DSDqOPFrVbbi/files",
    "watermark": {
      "url": "http://c3415543.r43.cf0.rackcdn.com/IEjQM3VBNkzI.jpg"
    }
  },
  "name": "New Player (Updated)",
  "publish_webhook": "",
  "studio_uid": "Ok82U2krWVzp",
  "studio_url": "http://app.ustudio.com/api/v2/studios/Ok82U2krWVzp",
  "tag": "player",
  "uid": "DSDqOPFrVbbi",
  "unpublish_webhook": "",
  "update_webhook": "",
  "videos_url": "/api/v2/studios/Ok82U2krWVzp/destinations/DSDqOPFrVbbi/videos"
}
```

The `files` entry now lists the `watermark` file with the URL of the uploaded file. Additional information may be provided as required by the Destination Template.

Deleting a Destination

Deleting a Destination is a very simple process. Note, however, that a Destination cannot be deleted unless all of the videos are unpublished. This is a safety measure, as once a Destination has been removed the information about any Videos published to it is also deleted, and cannot be retrieved later.

To delete a Destination, simply perform a `DELETE` request to the Destination URL. A successful response should return a `204` code, and will not contain any content.

Publishing a Video

Adding a video to a Destination requires a `video_uid`. This can be retrieved via the `videos_url` returned from the Studio resource, or from a Video created via the Upload API. (See the Upload API documentation for more details.) Also, the Destination must be successfully authenticated.

A `POST` to the `videos_url` (obtained from the Destination representation as previously documented) with the `video_uid` will begin the publishing process:

```
METHOD: POST
URL: https://app.ustudio.com/api/v2/studios/0k82U2krWVzp/destinations/DSDqOPFrVbbi/videos
REQUEST BODY:
{ "video_uid": "U06XcJJUFUOw" }
```

This will result in a `201` if the video has not been published to the Destination. If the Video has already been published, it will result in a `200` and no additional steps will be taken by the system, since a Video can only be published to a Destination once.

The response will contain the representation of the Video with any additional attributes available due to the Destination's capabilities. For instance, after publishing the previous video to the Player destination we created, the response looks like:

```
{
  "category": "business",
  "created": "Thu January 30 20:53:42 2013",
  "description": "This is a simple video promo.",
  "destination_video_url": "/api/v2/studios/0k82U2krWVzp/destinations/DSDqOPFrVbbi/videos/U06XcJJUFU0w",
  "failed": false,
  "image_url": "http://c3408616.r16.cf0.rackcdn.com/I5tpF13AGcMY.jpg",
  "oembed_url": "http://app.ustudio.com/api/v2/oembed?url=http%3A%2F%2Fapp.ustudio.com%2Fembed%2FDSDqOPFrVbbi%2FU06XcJJUFU0w",
  "player_embed_url": "http://app.ustudio.com/embed/DSDqOPFrVbbi/U06XcJJUFU0w",
  "published": true,
  "sharing_widget_url": "http://app.ustudio.com/static/api/js/player/sharing_widget.js",
  "status": "publish_succeeded",
  "title": "Promo Video",
  "uid": "U06XcJJUFU0w",
  "upload_url": "https://upload03-app.ustudio.com/api/v2/studios/0k82U2krWVzp/videos/U06XcJJUFU0w/asset",
  "video_template_tag": "standard",
  "video_template_uid": "CkQEeD3QrYED",
  "video_url": "/api/v2/studios/0k82U2krWVzp/videos/U06XcJJUFU0w",
  "web_page": "http://ustudio.com"
}
```

All Videos published to a Destination expose a few new attributes. The most important are `destination_video_url`, which will be used to manage the Video on the Destination, and both `published` and `failed`, which indicate the state of the publish operation.

This particular Video representation has also been augmented by a number of Player-specific attributes (`oembed_url`, `player_embed_url`, and `sharing_widget_url`). Based on the Destination, these serve different purposes and enable custom behavior. As an example, see the OEmbed section further in this document to see how a Player embed code can be generated simply from the `oembed_url` value.

Retrieving a List of Published Videos

To retrieve a list of Videos that are published (or being published) to a Destination, perform a `GET` request on the `videos_url` returned from the

Destination resource. The following is an example response with a single published Video:

```
{
  "count": 1,
  "destination_url": "/api/v2/studios/0k82U2krWVzp/destinations/DSDqOPFrVbbi"
,
  "studio_url": "/api/v2/studios/0k82U2krWVzp",
  "videos": [
    {
      "category": "business",
      "created": "Thu Jan 30 20:53:42 2013",
      "description": "This is a simple video promo.",
      "destination_video_url": "/api/v2/studios/0k82U2krWVzp/destination
s/DSDqOPFrVbbi/videos/U06XcJJUFU0w",
      "failed": false,
      "image_url": "http://c3408616.r16.cf0.rackcdn.com/I5tpFl3AGcMY.jpg"
    ,
      "oembed_url": "http://app.ustudio.com/api/v2/oembed?url=http%3A%2F
%2Fapp.ustudio.com%2Fembed%2FDSDqOPFrVbbi%2FU06XcJJUFU0w",
      "player_embed_url": "http://app.ustudio.com/embed/DSDqOPFrVbbi/U06
XcJJUFU0w",
      "published": true,
      "sharing_widget_url": "http://app.ustudio.com/static/api/js/player
/sharing_widget.js",
      "status": "publish_succeeded",
      "title": "Promo Video",
      "uid": "U06XcJJUFU0w",
      "upload_url": "https://upload02-app.ustudio.com/api/v2/studios/0k8
2U2krWVzp/videos/U06XcJJUFU0w/asset",
      "video_template_tag": "standard",
      "video_template_uid": "CkQEeD3QrYED",
      "video_url": "/api/v2/studios/0k82U2krWVzp/videos/U06XcJJUFU0w",
      "web_page": "http://ustudio.com"
    }
  ]
}
```

The Destination Video list is a paginated result, so it will only return the first 50 videos by default. You can specify an offset with the `index` query parameter, and a different page size with the `limit` parameter. For instance, a URL to grab videos 100 through 200 would look like:

```
https://app.ustudio.com/api/v2/studios/0k82U2krWVzp/destinations/DSDqOPFrVbbi/videos?limit=100&index=100
```

Finally, this resource is searchable. By providing a `query` parameter, only the Videos that a.) are published to this destination and b.) match the search term are returned. The search checks both title and keywords, but not the Video description at this time.

This example URL will return the first 10 published Videos that match the search term “video”:

```
https://app.ustudio.com/api/v2/studios/0k82U2krWVzp/destinations/DSDqOPFrVbbi/videos?limit=10&query=video
```

Unpublishing a Video

Unpublishing a Video, much like deleting a Destination, is a straightforward operation. Simply perform a `DELETE` request on the `destination_video_url` generated at publish time. A valid unpublish request will result in a `202`, indicating that the uStudio platform has started the unpublish process.

If the unpublish operations are unsuccessful (for example, if the Destination is in an unauthorized state, or there is an error with a third-party, etc.) the Video will not be removed from the Destination. After correcting the issue, the `DELETE` request will need to be resent.

Note that removing a Video from a Destination will irrevocably delete all publish information associated with that Video on that particular Destination.

Player-Specific OEmbed Example

As an example of what is possible with a specific Video published on a Destination, the following utilizes the `oembed_url` attribute returned from a Video published on a Player Destination.

By performing a `GET` request on the `oembed_url`, an application can easily obtain an optimized embed code for a player. Combined with appropriate caching, this protects the application from changing embed codes in the future without having to maintain lots of embed-specific code.

For instance, a simple `GET` request to the `oembed_url` for the Video we published in this document generates the following JSON response:

```
{
  "description": "This is a simple video promo.",
  "height": 360,
  "html": "<iframe src=\"http://app.ustudio.com/embed/DSDqOPFrVbbi/U06XcJJUFUOw\" width=\"640\" height=\"360\" frameborder=\"0\" scrolling=\"no\" allowFullScreen=\"true\" webkitAllowFullScreen=\"true\" mozAllowFullScreen=\"true\">
</iframe>",
  "provider_name": "uStudio",
  "provider_url": "http://ustudio.com",
  "thumbnail_height": 360,
  "thumbnail_url": "http://c3408616.r16.cf0.rackcdn.com/I5tpF13AGcMY.jpg",
  "thumbnail_width": 640,
  "title": "Promo Video",
  "type": "video",
  "version": "1.0",
  "width": 640
}
```

The key attributes here are `html`, `width`, and `height`. The `html` contents can be simply dropped into the HTML for a web site and the video will embed properly. If the space is constrained, `maxwidth` and `maxheight` values may be provided to ensure the player fits in the provided space. The following URL has been updated to force a player that fits into a 400px-wide area:

```
http://app.ustudio.com/api/v2/oembed?url=http%3A%2F%2Fapp.ustudio.com%2Fembed%2FSDSDqOPFrVbbi%2FU06XcJJUFUOw&maxwidth=400
```

The (simplified) response looks like:


```
{
  "html": "<iframe src=\"http://app.ustudio.com/embed/DSDqOPFrVbbi/
    U06XcJJUFU0w\" width=\"400\" height=\"225\" frameborder=\"0\"
    scrolling=\"no\" allowFullScreen=\"true\"
    webkitAllowFullScreen=\"true\" mozAllowFullScreen=\"true\">
    </iframe>\n",
  "width": 400,
  "height": 225
}
```

Note that the `height` value was updated to match the aspect ratio, even though `maxwidth` was the only value provided. For more information on OEmbed, visit the [specification page](#).