

Documentation

**Unpublished**

On This Page

Why migrate?
Migration paths
Video/Asset ingest
Client-facing Media API reads
Metadata writes
Video Field Comparison
API Methods Comparison
Related content

Migrating from the Legacy APIs to the Catalog APIs

[View \(/migrating-legacy-apis-catalog-apis\)](#)[Edit \(/node/17955/edit\)](#)[Delete \(/node/17955/delete\)](#)[Manage display \(/node/17955/manage-display\)](#)[Revisions \(/node/17955/revisions\)](#)

Video Cloud

API Developer

Legacy Migration, CMS API, Dynamic Ingest API, Playback API

This topic provides guidance on migrating your apps and Video Cloud integrations from the legacy Media API to the new Catalog APIs (Dynamic Ingest API, CMS API, and Playback API).

Why migrate?

There are several good reasons why you should migrate to the Catalog APIs:

- The Dynamic Ingest API provides pull-based or source file upload ingestion, with better performance and reliability.
- The CMS API provides cacheless read access, as well as functionality not available via the Media API.
- The Playback API provides fast access to video and playlist data, without exposing metadata that the player doesn't need.
- The new APIs are more scalable and reliable.
- The new APIs are being actively enhanced, while the Media API and Batch Provisioning are deprecated and will be discontinued in 2017.
- The Brightcove OAuth service provides a greater level of security for your content.

Migration paths

The diagram below shows the primary migration paths from the legacy APIs to the new APIs. The sections that follow provide more detail on the paths.

Client Side Media API Reads → Playback API

Media API Metadata Writes → CMS API

Media API/Batch Uploads → Dynamic Ingest API


Figure 1: Migration Paths

Video/Asset ingest

In the new Brightcove APIs, ingest functions are separate from media management functions - the rule to remember is that if you are sending files to Video Cloud, they will go via the Dynamic Ingest API (/overview-dynamic-ingest-api-legacy-ingest), **never** the CMS API. If you are currently using the Media API or Batch Provisioning to upload videos, images, or captions, you should now migrate that functionality to the Dynamic Ingest API.

Use cases

Here is a breakdown of some specific use cases:

Use Case	Migration Path
Using the Media API to upload videos from your CMS	Use the Dynamic Ingest API (/overview-dynamic-ingest-api-legacy-ingest) instead
Using Batch Provisioning to upload videos	Use the Dynamic Ingest API - you can host the files on your own server or S3 bucket for pull-based ingest, or use source file upload (/source-file-upload-api-dynamic-ingest) ^[1]  If you will be ingesting large numbers of videos, also see this guide (/fallback-retry-strategy) to using Notifications to avoid exceeding ingest rate limits.
Using Batch Provisioning to replace or retranscode videos	Both of these operations can be performed using the Dynamic Ingest API (/overview-dynamic-ingest-api-legacy-ingest#replaceVideo) - you can host the files on your own server or S3 bucket for pull-based ingest, or use source file upload (/source-file-upload-api-dynamic-ingest) ^[2]
Using the Media API or Batch Provisioning to add images or captions to videos	Video stills (posters) and thumbnails (/images-and-dynamic-ingest-api) and WebVTT text tracks (/ingesting-webvtt-files-text-tracks) can be added using the Dynamic Ingest API. ^[3]

Notes

- ^[1] The Dynamic Ingest API does not support multiple ingest on one request, but you can iterate over a set of video data as in this example (/dynamic-ingest-api-sample-ingesting-videos-legacy-ingest)
- ^[2] Again, each video requires a separate request, but you can iterate over data sets as in these samples:
 - Replace Multiple Videos (/dynamic-ingest-api-sample-replacing-videos-legacy-ingest)
 - Re-transcode Multiple Videos (/dynamic-ingest-api-sample-re-transcoding-videos-legacy-ingest)
- ^[3]



Other media assets

Logo overlays

Logo overlays will be handled differently:

- For burned-in logos, you can specify a watermark (/ingest-profiles-api-fields-reference-legacy-ingest#watermarkFields), which is an image inserted into the video data during transcoding (just as the old burn-in logos were). Watermarks will travel with the video renditions even if they are played in non-Brightcove players (but not when the videos are distributed to social media via Brightcove Social, because we always send the digital master to social media providers).
- For logo overlays, use the logo overlay plugin (/brightcove-player-sample-logo-overlay-plugin).

DFXP captions

DFXP captions are used only with the legacy smart player - if you are still using the smart player and need to add captions to videos, you can host the DFXP files yourself and add them to videos using the CMS API (https://docs.brightcove.com/cms-api/v1/doc/index.html#api-assetGroup-Add_Caption).

Client-facing Media API reads

If you are currently using the Media API to pull video or playlist data from Video Cloud you should migrate this functionality to use the Playback API (/overview-playback-api) instead. For web pages that contain a player, there is no need to make requests to the Playback API directly, as this functionality is encapsulated in the player catalog (/player-catalog).

For mobile apps or web pages that don't contain a player, such as video portal or catalog, you can submit requests to the Playback API directly, using a policy key (/overview-policy-api) for authentication.



If you just need a basic policy key, you can use the app here (/quick-start-policy-api) instead of making the Policy API request yourself.

Use cases

Here is a breakdown of some specific use cases:

Current State	Migration Path
Using the Media API to retrieve videos or a playlist by ID to load into the Brightcove Player dynamically	Use the player catalog (/player-catalog) methods, iterating if you need to fetch multiple items.
Using the Media API to retrieve videos or a playlist in an iOS or Android app	Use the Playback API (/overview-playback-api). The Native Player SDKs offer endpoints to facilitate getting media metadata - see Working with Media Content (iOS) (/ios-working-media-content) and Working with Media Content (Android) (/android-working-media-content) for more details.
Using the Media API to search for videos (related to the current video, etc.)	Use the Playback API search (/cmsplayback-api-videos-search)
Using the Media API <code>find_related_videos</code> to get videos related to the one in the player (either directly or via the Smart Player settings)	See the following samples: <ul style="list-style-type: none"> Related Videos from a Playlist (/brightcove-player-sample-related-videos-playlist) Related Videos via Tags (/brightcove-player-sample-related-videos-tags-cms-api) (note that this sample could also use the Playback API for a solution that does not require a server-side proxy)

Metadata writes


Write requests to the Video Cloud system should always be made on the server side (if you need a web UI, make the requests via a server-side proxy that holds the authentication credentials). Where you are using the Media API to manage video metadata, including adding remote assets, you should migrate this functionality to the CMS API (/overview-cms-api). To add videos or other assets to Video Cloud, see the next section.

See the API methods comparison below to see equivalent CMS API methods for various Media API methods, and the Video fields comparison to see where the names of video properties differ between the Media and CMS APIs.

Use cases

Here is a breakdown of some specific use cases:

Current State	Migration Path
Using the Media API to update video metadata (name, description, tags, etc.)	Use the CMS API Update Video (/docs.brightcove.com/cms-api/v1/doc/index.html#api-videoGroup-Update_Video) operation.
Using the Media API to manage playlists	Use the CMS Playlist operations (/docs.brightcove.com/cms-api/v1/doc/index.html#api-playlistGroup).
Using Batch Provisioning to update video metadata	Use the CMS API Update Video (/docs.brightcove.com/cms-api/v1/doc/index.html#api-videoGroup-Update_Video) method
Using Batch Provisioning to apply geo-filtering to videos	Use the CMS API Update Video (/docs.brightcove.com/cms-api/v1/doc/index.html#api-videoGroup-Update_Video) method ^[4]

Using Media API or Batch Provisioning to share videos between accounts	 Support for Media Sharing is planned but not yet available in the CMS API - you will need to wait until this feature is added.
Using Batch Provisioning to retranscode videos	Use the Dynamic Ingest API (/overview-dynamic-ingest-api-legacy-ingest#retranscode) ^[5]
Using Batch Provisioning or the Media API to add remote remote videos and assets such as captions and images	Use the CMS API assets operations (/docs.brightcove.com/cms-api/v1/doc/index.html#api-assetGroup) - also see Using the CMS API: Remote Assets (/using-cms-api-remote-assets)

In short, the new Catalog APIs offer greater performance, reliability, and functionality. In addition, they are part of the larger platform that is the new Video Cloud - in addition to the APIs discussed here, there are APIs for Analytics, Player Management, Ingest Profiles, and OAuth. Putting these all together enables you to create a complete integration between your CMS or app and Video Cloud.

Notes

- ^[4] The CMS API does not support updates on multiple videos - iterate over a video data set, making an Update Video (/docs.brightcove.com/cms-api/v1/doc/index.html#api-videoGroup-Update_Video) request for each video - see a sample app here (/cms-api-sample-update-geo-properties).
- ^[5] Dynamic Ingest does not support processing multiple videos in one request, but you iterate over a data set to handle multiple videos, as in this sample (/dynamic-ingest-api-sample-re-transcoding-videos-legacy-ingest)

Video Field Comparison

The table below lists video fields that are different in CMS API responses from those in the Media API. Note that only those fields that are different are listed - video fields (/cms-api-custom-fields-reference) not listed are the same for both APIs.

Media API Field	CMS API Field	Type	Can be updated in CMS API?	Description
accountId	account_id	String	no	
adKeys	ad_keys	String	yes	key/value pairs for ad requests
captioning	text_tracks	Array of HTML5-style text tracks	yes	WebVTT assets; more details here (/using-cms-api-add-webvtt-captions-text-tracks)
	id	String	no	
	src	String	yes	address of the track file (URL)
	srcLang	String	yes	language of the track, "en"
	label	String	yes	a user-readable title
	kind	String	yes	how the track is meant to be used: "subtitles", "captions", "descriptive", "chapters", "metadata"
	mime_type	String	yes	mime type of the track e.g. "text/vtt"

	<i>id_asset</i>	String	no	null if asset is remote
	<i>sources</i>	Array	no	
	<i>src</i>	String	yes	address of the track file (URL)
	<i>default</i>	Boolean	yes	indicates that the track should be enabled by default. Should only be on one track.
creationDate	created_at	UTC date-time	no	
cuePoints	cue_points	Array of Maps	yes	
	<i>cue_points.0.id</i>	String	yes	
	<i>cue_points.0.force_stop</i>	Boolean	yes	
	<i>cue_points.0.metadata</i>	_String	yes	code point only_
	<i>cue_points.0.name</i>	String	yes	
	<i>cue_points.0.time</i>	Float	yes	
	<i>cue_points.0.type</i>	String	yes	
customFields	custom_fields	Map of field-value pairs (Strings)	yes	
dashManifestUrl	dash_manifest_id	String	no	Was an URL, is now a
dashRenditions	n/a			See sources (https://docs.brightcove.com/api/v1/doc/index.html#videoGroup-Get_Video_Sources)
digitalMaster	<i>digital_master_id</i>	String	no	Was an object; is now a
economics	economics	String enum	yes	FREE, AD_SUPPORTI
FLVURL	n/a			See sources (https://docs.brightcove.com/api/v1/doc/index.html#videoGroup-Get_Video_Sources)
	geo	Map of property-value pairs	yes	
geoFilteredCountries	<i>geo.countries</i>	Array of country code Strings	yes	
geoFilterExclude	<i>geo.exclude_countries</i>	Boolean	yes	
geoRestricted	<i>geo.restricted</i>	Boolean	yes	
geoFiltered	n/a			
hdsManifestUrl	n/a			See sources (https://docs.brightcove.com/api/v1/doc/index.html#videoGroup-Get_Video_Sources)

				videoGroup-Get_Video_Sources)
HDSRenditions	n/a			See sources (http://docs.brightcove.com/api/v1/doc/index.htm#videoGroup-Get_Video_Sources)
HLSURL	n/a			See sources (http://docs.brightcove.com/api/v1/doc/index.htm#videoGroup-Get_Video_Sources)
id	id	String	no	Was a Long, is now a
IOSRenditions	n/a			See sources (http://docs.brightcove.com/api/v1/doc/index.htm#videoGroup-Get_Video_Sources)
itemState	state	String enum	yes	ACTIVE, INACTIVE, PENDING, DELETED
lastModifiedDate	updated_at	UTC date-time	no	
length	duration	Number, milliseconds	no	
	link	Map of property-value pairs	yes	
linkUrl	<i>link.url</i>	String	yes	
linkText	<i>link.text</i>	String	yes	
logoOverlay	n/a			
longDescription	long_description	String	yes	
name	name	String	yes	
playsTotal	n/a			
playsTrailingWeek	n/a			
publishedDate	n/a			
referenceId	reference_id	String	yes	
renditions	n/a			See sources (http://docs.brightcove.com/api/v1/doc/index.htm#videoGroup-Get_Video_Sources)
	schedule	Map of property-value pairs	yes	
endDate	<i>schedule.ends_at</i>	String in ISO-8601 date format	yes	
startDate	<i>schedule.starts_at</i>	String in ISO-8601	yes	

		<i>date format</i>		
	sharing	Map of property-value pairs	no	
sharedByExternalAccount	<i>sharing.by_external_account</i>	Boolean	no	
	<i>sharing.by_id</i>	String	no	pub ID that originates share
	<i>sharing.source_id</i>	String	no	video ID of the origin video (sharer's copy)
sharedToExternalAccount	<i>sharing.toexternal acct</i>	Boolean	no	
	<i>sharing.by_reference</i>	Boolean	no	
shortDescription	description	String	yes	
	<i>smooth_client_manifest_id</i>	String	no	
smoothManifestUrl	<i>smooth_server_manifest_id</i>	String	no	Was a URL, is now an
smoothRenditions	n/a			See sources (//docs.brightcove.com/api/v1/doc/index.html#videoGroup-Get_Video_Sources)
tags	tags	Array of tags (Strings)	yes	
	images	Map of property-value pairs	no	
thumbnail	<i>images.thumbnail</i>	Map	no	
	<i>images.thumbnail.asset_id</i>	String	no	
	<i>images.thumbnail.src</i>	String URL	no	
	<i>images.thumbnail.sources</i>	Array of Maps	no	
	<i>images.thumbnail.sources.asset_id</i>	String	no	
thumbnailUrl	<i>images.thumbnail.sources.src</i>	String URL	no	
	<i>images.thumbnail.sources.height</i>	Number	no	
	<i>images.thumbnail.sources.width</i>	Number	no	
videoStill	<i>images.poster</i>	Map	no	
	<i>images.poster.asset_id</i>	String	no	
	<i>images.poster.src</i>	String URL	no	
	<i>images.poster.sources</i>	Array of Maps	no	
	<i>images.poster.sources.asset_id</i>	String	no	
videoStillUrl	<i>images.poster.sources.src</i>	String URL	no	
	<i>images.poster.sources.height</i>	Number	no	

	<i>images.poster.sources.width</i>	Number	no	
	original_filename	String	no	Will be null for remot videos
version	n/a			See the etag header c request reponse
videoFullLength	n/a			See sources (//docs.brightcove.com/api/v1/doc/index.htm#videoGroup-Get_Video_Sources)
WMVRenditions	n/a			See sources (//docs.brightcove.com/api/v1/doc/index.htm#videoGroup-Get_Video_Sources)
n/a	complete	Boolean	no	
n/a	folder_id	String	no	Mongo-type ID of fol which the video belo

API Methods Comparison

The table below shows Media API methods and the equivalent operation in CMS (or Dynamic Ingest) API

Media API	CMS API Verb + Endpoint
Video Methods	
search_videos	GET /accounts/account_id/videos + search (/cmsplayback-api-videos-search)
find_modified_videos	GET /accounts/{account_id}/videos?sort=%2DUpdated_at
find_video_by_id	GET /accounts/{account_id}/videos/{video_id}
find_videos_by_ids	GET /accounts/{account_id}/videos/{video_id} (iterate over ids)
find_video_by_reference_id	GET /accounts/{account_id}/videos/ref:{reference_id}
find_videos_by_reference_ids	GET /accounts/{account_id}/videos/ref:{reference_id} (iterate over reference ids)
find_all_videos	GET /accounts/account_id/videos
find_related_videos	GET /accounts/account_id/videos + search (/cmsplayback-api-videos-search) (search according to the relationships you want)
create_video	POST /accounts/account_id/videos + Dynamic Ingest API (/overview-dynamic-ingest-api-legacy-ingest)
update_video	PATCH /accounts/{account_id}/videos/{video_id}
delete_video	DELETE /accounts/{account_id}/videos/{video_id}
add_captioning	Use Dynamic Ingest - Text Tracks (/ingesting-webvtt-files-text-tracks)
add_image	Dynamic Ingest - Images (/images-and-dynamic-ingest-api)
add_logo_overlay	Either use the Player Overlay Plugin (/display-overlay-plugin) for detached overlays that can be managed and changed at runtime, or burned-in logos through a custom Ingest Profile including watermarks (/ingest-profiles-api-fields-reference-legacy-ingest#watermarkFields)

delete_captioning	PATCH /accounts/{account_id}/videos/{video_id}
get_upload_status	GET /accounts/{account_id}/videos/{video_id} (status is included in the response)
remove_logo_overlay	No equivalent - see notes on add_logo_overlay above
share_video	see Sharing Videos with the CMS API (/node/19325)
unshare_video	see Sharing Videos with the CMS API (/node/19325)
Playlist Methods	
find_all_playlists	GET /account/{account_id}/playlists
find_playlist_by_id	GET /account/{account_id}/playlists/{playlist_id}
find_playlist_by_reference_id	GET /account/{account_id}/playlists/ref:{reference_id}
find_playlists_by_ids	GET /account/{account_id}/playlists/{playlist_id} (iterate over ids)
find_playlists_by_reference_ids	GET /account/{account_id}/playlists/ref:{reference_id} (iterate over reference ids)
find_playlists_for_player_id	no equivalent
create_playlist	POST /account/{account_id}/playlists
update_playlist	PATCH /account/{account_id}/playlists/{playlist_id}
delete_playlist	DELETE /account/{account_id}/playlists/{playlist_id}

Related content

[Viewing the Brightcove System Status Page \(/viewing-brightcove-system-status-page\)](#)
[Brightcove Player Migration Guide \(/brightcove-player-migration-guide\)](#)
[Publishing Videos and Playlists with the Media Module \(/publishing-videos-and-playlists-media-module\)](#)
[Getting Started with the Media Module \(/getting-started-media-module\)](#)
[Getting Started with the Players Module \(/getting-started-players-module\)](#)
[Working with Remote Asset Videos \(/working-remote-asset-videos\)](#)
[Setting Player Properties \(/setting-player-properties\)](#)
[Creating and Managing Playlists \(/creating-and-managing-playlists\)](#)
[Creating Custom Lead Forms for Salesforce \(/creating-custom-lead-forms-salesforce\)](#)
[Quick Start: Using the Players Module to Create and Publish a Player \(/quick-start-using-players-module-create-and-publish-player\)](#)

How helpful was this topic?

☐ Very Helpful ☐ ☐ ☐ Not Helpful

Brightcove Support

[System Status](#)
<https://status.brightcove.com>
[Open a Support Case](#)
[\(/contact-support\)](#)

Documentation

[Home \(/\)](#)

Training

[Online Courses \(/online-training-courses\)](#)
[Register for a course](#)
[\(/register-training\)](#)

Brightcove

[Brightcove.com](#)
<https://www.brightcove.com>
[Contact Us \(/contact-us\)](#)

Brightcove Inc. (NASDAQ: BCOV), is a leading global provider of powerful cloud solutions for delivering and monetizing video across connected devices.

(https://www.brightcove.com/learning/)

© 2018 Brightcove Inc. All rights reserved.
| Privacy (https://www.brightcove.com/en/privacy)
| Terms & Conditions (https://www.brightcove.com/en/terms-and-conditions)