Dynamic Download and File Protection

Dynamic File Download

Files can now be downloaded using dynamic URLs.

With dynamic URLs, you don't need the API to find the exact download URL for an unprotected file (described in the next section).

Instead, you can locate the file if you know the asset type's URL scheme and have information about the asset, such as its ID.

If the file is protected, you have to look up the download URL to have a valid token.

Additionally, you would need to re-check the URL every time the token expires (they are valid for 10 minutes).

The following table provides the dynamic URL scheme for each asset type:

Dynamic File Downloading Scheme		Notes
Video Source Files	`{DOMAIN}/public/videos/ {VIDEO_ID}/source`	
Video Format Files	`{DOMAIN}/public/videos/ {VIDEO_ID}/formats/{PROFILE_KEY}`	
Thumbnails	`{DOMAIN}/public/videos/ {VIDEO_ID}/thumbnails/active? quality={QUALITY(e.g. 720p)}`	Note the options here: you can specify either the active thumbnail, or specify the thumbnail ID.
	`{DOMAIN}/public/videos/ {VIDEO_ID}/thumbnails/ {THUMBNAIL_ID}?quality={QUALITY (e.g. 720p)}`	The quality parameter is optional (it defaults to 360p).
Chapter Thumbnails	`{DOMAIN}/public/videos/ {VIDEO_ID}/chapters/{CHAPTER_ID} /thumbnails?quality={QUALITY(e. g. 720p)}`	The quality parameter is optional (it defaults to 360p).
Subtitles	`{DOMAIN}/public/videos/ {VIDEO_ID}/subtitles/ {SUBTITLE_ID}?source=VTT`	The "source" value indicates the subtitle file format. Possible values include: "SRT" or "VTT".
Video Attachments	`{DOMAIN}/public/videos/ {VIDEO_ID}/attachments/ {TYPE_NAME}/{FILENAME}`	The attachment types are set up in the admin area of VideoManager Pro. Use the name of the type here, not the ID. The filename is the name of the file that was uploaded. If these values contain special characters or spaces, they should be URL encoded.
Channel Attachments	`{DOMAIN}/public/channels/ {CHANNEL_ID}/attachments/ {TYPE_NAME}/{FILENAME}`	The attachment types are set up in the admin area of VideoManager Pro. Use the name of the type here, not the ID. The filename is the name of the file that was uploaded. If these values contain special characters or spaces, they should be URL encoded.

ⓘ Files on the general instance of the movingimage Platform use the following domain: "https://asset-out-cdn.video-cdn.net".

File Protection

File protection refers to a user's ability to access files on the movingimage Platform. When the user is logged into the Platform, the ownership features of VideoManager Pro control this. The logged in user can access files according to their rights. The new feature in this beta, however, concerns the protection of files from users who are not logged into the Platform.

By default, files are protected from non-logged-in users via the use of tokens provided in the file's download URL (tokens are valid for 10 minutes). The download URLs can be acquired from the movingimage Platform API. The following is an example download URL for a protected/private video thumbnail file:

```
https://asset-out-cdn.video-cdn.net/private/videos/-civ9qE3JUjntKQM1HnGtU/thumbnails/78912?
quality=360p&__token__=exp=1519826701~acl=***~hmac=***
```

File protection of this variety can now be disabled if desired. This would allow the file to be accessible on the public domain, without a token:

```
https://asset-out-cdn.video-cdn.net/public/videos/-civ9qE3JUjntKQM1HnGtU/thumbnails/78912?quality=360p
```

Note that the unprotected file specifies "/public/" in the URL instead of "/private/", and does not include a token.

All files are protected by default and only a user with admin rights in VideoManager Pro can disable file protection. Unpublished files are only accessible via VideoManager Pro or the movingimage Platform API. File protection can be disabled either for all assets within a VideoManager, or for a video or channel's specific asset type (i.e. thumbnail, chapter thumbnail, attachment, subtitle, etc.).

Use the following API endpoints to disable file protection:

VideoManager Level Requests

ⓘ You can also find the interactive API documentation here.

Endpoint:	/v1/vms/{videoManagerId}/security-override

This endpoint allows you to get or modify file protection for all files within a VideoManager.

HTTP Method	Request Header	Request Body	Response Body	Description

GET	Authorization : Bearer "{token}"	-	Example 1: All files are protected (scope is "private", and no "public" override exists for videos, channels or asset types):
			
			``
			{
			    "scope": "private",
			    "publicAssetTypes": [],
			    "videos": {},
			    "channels": {}
			}
			``
			
			Example 2: All files are in general protected, but the video source and format files for one video are public, and all asset types of one channel are public:
			
			``
			{
			    "scope": "private",
			    "publicAssetTypes": [],
			    "videos": {
			        "EYk8-jADvHnGqgwcyLJUNh": {
			            "assetTypes": [
			"VIDEO_SOURCE",
			"VIDEO_FORMAT"
			            ],
			            "scope":
			"public"
			        }
			    },
			    "channels": {
			        "314159": {
			            "assetTypes":
			null,
			            "scope":
			"public"
			        }
			    }
			}
			``
				Get file protection details for the specified VideoManager. The VideoManager scope (public or private) will be returned along with a list of video and/or channel assets that have been made public (if any). See the example responses to the left for details.

POST	Authorization : Bearer "{token}"	-	-	This request makes all assets within the specified VideoManager public. You do not need to include a request body.

DELETE	Authorization : Bearer "{token}"	-	-	This request makes all assets in the VideoManager private/protected again (including all Channel and Video assets).

Asset Level Requests

ⓘ You can also find the interactive API documentation here.

Endpoint:	/v1/vms/{videoManagerId}/asset-type/security-override

This endpoint allows you to remove file protection for specific asset types at the VideoManager level.

HTTP Method	Request Header	Request Body	Description

POST	Authorization: Bearer "{token}"	```
		{
		    "assetTypes": [
		"VIDEO_SOURCE",
		"VIDEO_FORMAT",
		"STILL_SOURCE",
		"STILL_RESOLUTION",
		    "SUBTITLE",
		    "ATTACHMENT"
		  ]
		}
		``
			This request can be used to specify which asset types are to be made public for the whole VideoManager.
			
			All possible asset types are listed in the example. You must specify at least one asset type.

Channel Level Requests

Endpoint:	/v1/vms/{videoManagerId}/channels/{channelId}/security-override

This endpoint allows you to remove file protection for only one asset type, **attachment**, at the Channel level.

HTTP Method	Request Header	Request Body	Description

POST	Authorization: Bearer "{token}"	```
		{
		"assetTypes": [
		"ATTACHMENT"
		  ]
		}
		``
			This request can be used to specify which asset types are to be made public within the hannel.
			
			ⓘ Channels currently only support the **attachment** asset type.

DELETE	Authorization: Bearer "{token}"	-	Use this request to make all assets in the specified Channel private/protected again.

Video Level Requests

ⓘ You can also find the interactive API documentation here.

Endpoint:	/v1/vms/{videoManagerId}/videos/{videoId}/security-override

This endpoint allows you to remove file protection for assets at the Video level. Note that there are two possible POST requests which work differently (see details in the table below):

HTTP Method	Request Header	Request Body	Description
POST	Authorization: Bearer "{token}"	`{ "scope": "public" }`	This request makes all assets within the specified Video public.

POST	Authorization: Bearer "{token}"	`{`
		`  "assetTypes": [`
		`"VIDEO_SOURCE",`
		`"VIDEO_FORMAT",`
		`"STILL_SOURCE",`
		`"STILL_RESOLUTION",`
		`    "SUBTITLE",`
		`    "ATTACHMENT"`
		`  ]`
		`}`
			This request can be used to specify which asset types are to be made public within the Video.
			
			ⓘ All possible asset types are listed in the example. You must specify at least one asset type.

DELETE	Authorization: Bearer "{token}"	-	Use this request to make all assets in the specified Video private/protected again.