# How to retrieve the download or streaming URL using API calls

Last Modified on 09/15/2025 12:44 pm IDT

This article is designated for administrators.

The Kaltura Player abstracts the need to retrieve direct access to the video file, and handles the various aspects of the video playback including multi-bitrate, choosing the correct codec and streaming protocols, DRM, Access Control and more. Often applications need a direct URL for downloading the raw media file for purposes such as streaming outside of the Kaltura Player, processing the video data (such as video analysis or transcriptions) and more. In cases where you need to access the playback stream directly, or just a link to download the video file, you will need to consider the target playback devices, any security protocols applied to your Kaltura account and video, and then call the suitable API methods. The following will guide developers of using the playManifest API call to retrieve specific transcoded video flavors from a Kaltura account.

## The playManifest API

The playManifest is a redirect action [source code], its purpose is to direct media player applications to the desired video stream.

playManifest features return two types:

1. A redirect to video file for progressive download, or an M3U8 stream descriptor for Apple HTTP Streaming, aka HLS.
2. An XML response in the form of a MPEG-DASH Media Presentation Description (MPD).
3. An XML response in the form of Flash Media Manifest File.

### Retrieving a URL for a video stream

To retrieve a specific video flavor, call the **playManifest** API. Be certain to have the Partner ID and Entry ID at hand. To call the **playManifest** API and retrieve a specific video flavor, call the following URL -

**[serviceUrl]**/p/**[YourPartnerId]**/sp/0/playManifest/entryId/**[YourEntryId]**/format/**[StreamingFormat]**/protocol/**[Protocol]**/flavorPara

Replace the following parameters:

- **serviceUrl -** the base URL to the Kaltura Server
- **YourPartnerId** - Your Kaltura account publisher Id. (Can be retrieved from the Publisher Account Settings page in the KMC).
- **YourEntryId** - The Id of the media entry you'd like to retrieve.
- **StreamingFormat** - See the list of available formats in the table below. This parameter is optional and defaults to /format/url
- **Protocol** - Whether video is to be delivered over HTTP or HTTPS. See the list of available protocols below for additional options. This parameter is optional and defaults to /format/http
- **VideoFlavorId -** The Id of the video flavor you want to download. If supported by the streaming format, multiple flavors may be comma-separated.
- **ks** - A valid Kaltura Session. This parameter is only required when the media entry has an Access Control defined to limit anonymous access to the media.
- **ext - The file extension of the video you wish to retrieve (For example, mp4, if the video flavor is an MPEG4 file or flv, if the video flavor is an FLV file.)**

**Available service URLs (Public / SaaS)**

| Protocol + Domain | Description |
|---|---|
| https://cdnapisec.kaltura.com | Secure HTTPS Request. (recommended) |
| http://cdnapi.kaltura.com | Standard HTTP Request. |

**Available playback formats**

| Format | Description |
|---|---|
| mpegdash | MPEG-DASH Streaming. |
| applehttp | HTTP Live Streaming (HLS) Streaming. |
| url | Progressive download. |
| hds | Adobe HTTP Dynamic Streaming. *Not available for all Kaltura accounts.* |
| rtsp* | Real Time Streaming Protocol (RTSP). For legacy devices, such as older Blackberry and Nokia phones.*<br><br>⚠ *RTSP was deprecated. RTMP & RTMPS are substitutes. |
| hdnetworkmanifest | Akamai HDS delivery. *Available only for accounts with Akamai delivery. (deprecated)* |
| hdnetwork | Akamai Proprietary Delivery Protocol. *Available only for accounts with Akamai delivery.* **(deprecated)** |

**Available Protocol Parameters**

| Protocol | Description |
|---|---|
| http | http Redirect and streaming URLs make use of the HTTP protocol. (Default) |
| https | https  Redirect and streaming URLs make use of the HTTPS protocol. |

Examples:

- https://cdnapisec.kaltura.com/p/309/sp/0/playManifest/entryId/1_rcit0qgs/format/url/flavorParamId/301971/video.mp4
- https://cdnapisec.kaltura.com/p/309/sp/0/playManifest/entryId/1_rcit0qgs/format/applehttp/protocol/http/flavorParamId/301971/manifest.m3u8

### Considerations of access control and entitlements

It is important to note that Kaltura entries can be set for private or protected modes, where access is only allowed when providing a valid admin Kaltura Session.

For best practice, to retrieve the download URL for an entry, use the following steps:

1. Locate the Id of the desired video flavor (see below Video Flavor Id).
2. Call the flavorAsset.geturl API action.

Below is a PHP code sample for retrieving the download URL of a web-playable flavor for a desired entry Id:

```
//Client library configuration and instantiation...

//when creating the Kaltura Session it is important to specify that this KS should bypass entitlemets restrictions:
$ks = $client->session->start($secret, $userId, KalturaSessionType::ADMIN, $partnerId, 86400, 'disableentitlement');
$client->setKs($ks);

$client->startMultiRequest();
$entryId = '1_u7aj9kasw'; //replace this with your entry Id
$client->flavorAsset->getwebplayablebyentryid($entryId);
$req1ResultFlavorId = '{1:result:0:id}'; //get the first flavor from the result of getwebplayablebyentryid
$client->flavorAsset->geturl($req1ResultFlavorId); //this action will return a valid download URL
$multiRequestResults = $client->doMultiRequest();
$downloadUrl = $multiRequestResults[1];
echo 'The entry download URL is: '.$downloadUrl;
```

### Video flavor Id

The VideoFlavorId parameter determines which video flavor the API will return as download. This parameter has various options, depending on the Kaltura server deployment and publisher account.

The following lists few of the conventional flavor Id's:

Note: Only flavor id 0 (zero) is static and the same across Kaltura editions. The following list are common flavor Ids on the Kaltura SaaS edition, but note flavors change and upgraded often (improved quality, new codecs, etc.) - Use this list for example purposes.

- The original uploaded video (before transcoding) = 0
- iPhone / Android (mp4) = 301951
- iPad (mp4) = 301971
- Nokia/Blackberry (3gp) = 301991
- Other devices (mp4) = 301961

The correct flavor Ids (per account and Kaltura edition) can be retrieved by:

1. Visiting the KMC Settings > Transcoding Profiles.
2. OR by making an API call the ConversionProfile.list action.

### Retrieving streaming URL for mobile applications

To retrieve streaming URL for mobile applications, use the following guidelines:

For Apple iPad devices – get all the flavors (marked ready) that have the tag 'ipadnew' and build the following URL:

```
serviceUrl + '/p/' + partnerId + '/sp/' + partnerId + '00/playManifest/entryId/' + entryId + '/flavorIds/' + flavorIds.join(',') + '/format/applehttp/protocol/http/a.m3u8?ks=' + ks + '&referrer=' + base64_encode(application_name)
```

For Apple iPhone devices – get all the flavors (marked ready) that have the tag 'iphonenew' tag and build the following URL:

```
serviceUrl + '/p/' + partnerId + '/sp/' + partnerId + '00/playManifest/entryId/' + entryId + '/flavorIds/' + flavorIds.join(',') + '/format/applehttp/protocol/http/a.m3u8?ks=' + ks + '&referrer=' + base64_encode(application_name)
```

For Android devices that support HLS – get all the flavors (marked ready) that have the tag 'iphonenew' tag (excluding audio-only flavors where width, height & framerate fields equal to zero) and build the following URL:

```
serviceUrl + '/p/' + partnerId + '/sp/' + partnerId + '00/playManifest/entryId/' + entryId + '/flavorIds/' + flavorIds.join(',') + '/format/applehttp/protocol/http/a.m3u8?ks=' + ks + '&referrer=' + base64_encode(application_name)
```

For Android devices that do not support HLS – get a single video flavor that has the 'iphonenew' tag and build the following URL:

```
serviceUrl + '/p/' + partnerId + '/sp/' + partnerId + '00/playManifest/entryId/' + entryId + '/flavorId/' + flavorId + '/format/url/protocol/http/a.mp4?ks=' + ks + '&referrer=' + base64_encode(application_name)
```