NOTE: This PDF document was produced from a live HTML article. The PDF formatting does not fully reflect the live implementation.

Wowza Streaming Cloud REST API migration guide

Originally Published on 01/31/2020 | Updated on 01/31/2020 9:39 am PST

As you move the codebase for your streaming workflow from one version of the Wowza Streaming Cloud™ REST API to a later version, use this guide to track what's changed and how these changes affect your code.

Contents

Migrate from version 1.0 to 1.3

Migrating your existing Wowza Streaming Cloud integrations to version 1.3 of the REST API will allow you to take advantage of new features, performance improvements, and enhanced security.

Versions 1.0, 1.1, and 1.2 have been sunset, so this guide focuses on migrating from v1.0 to v1.3 to include all changes that a user migrating to v1.3 needs to be aware of.

A number of things have changed between API versions. While this guide includes most changes you need to know, see the Wowza Streaming Cloud REST API reference documentation (/help/wowza-streaming-cloud-rest-api-reference-documentation) to confirm specific details of endpoint behavior in v1.3.

Developer checklist

- Update your authentication method to HMAC signed authentication.
- Familiarize yourself with the new workflow for retrieving detailed information about individual objects - including transcoders, live streams, players, recordings, stream sources and stream targets of all types.
- Understand updates to stream target endpoint syntax and object references.
- Understand updates to stream source endpoint syntax and object references.
- Note pagination updates.
- Review deprecations to determine whether your code uses deprecated endpoints or parameters.
- Review added and updated functionality to determine whether you want to make use of new parameters or parameter values.
- Update your code as necessary based on changes to operations and objects from v1.0 to v1.3.

What's changed

Over the course of development, we made the following changes to the REST API from version 1.0 to version 1.3. Use the tables to compare workflows and endpoints in v1.0 to v1.3.

Authentication method

In v1.3, we added a more secure authentication method for use in production environments: hash-based message authentication code (HMAC). In this form of authentication, the API key is a private, secret key. It is known to you and the Wowza Streaming Cloud service but never sent directly in an API request. Headers for HMAC include an access key, a timestamp, and a signature generated using the HMAC-256-Hexdigest algorithm. See Authentication (how-to-use-the-wowza-streaming-cloud-rest-api#authentication) to learn how to set this up.

<table>
<tr><th>v1.0</th><th>v1.3</th></tr>
<tr>
<td>API key and access key authentication only
Copy
<pre>
curl -X GET \
-H "wsc-api-key: [your API key]" \
-H "wsc-access-key: [your access key]" \
"https://api.cloud.wowza.com/api/[version]/live_s
</pre>
</td>
<td>HMAC authentication for production environments
Copy
<pre>
curl -X GET \
-H "wsc-access-key: [your access key]" \
-H "wsc-timestamp: [timestamp]" \
-H "wsc-signature: [code-generated-signature]" \
"https://api.cloud.wowza.com/api/[version]/live_s
</pre>
API key and access key authentication for initial testing of API only
Copy
<pre>
curl -X GET \
-H "wsc-api-key: [your API key]" \
-H "wsc-access-key: [your access key]" \
"https://api.cloud.wowza.com/api/[version]/live_s
</pre>
</td>
</tr>
</table>

Retrieving detailed information about individual objects

In v1.3, we reduced the response information returned when you fetch the index of an object. This applies to GET requests for all transcoders, live streams, players, recordings, stream targets, Wowza stream targets, custom stream targets, ultra low latency stream targets, stream sources, and Akamai stream sources. Previously, responses included detailed information about each object. If you need to request detailed information about an object in v1.3, you'll need to fetch the index, then fetch a single object to view its details. See the v1.3 Wowza Streaming Cloud REST API reference documentation to see the specifics for each request.

You can use query parameters to limit the scope of your initial request. See Get paginated query results with the Wowza Streaming Cloud REST API (/help/how-to-get-paginated-query-results-with-the-wowza-streaming-cloud-rest-api) and Get paginated query results with the Wowza Streaming Cloud REST API (/help/how-to-get-paginated-query-results-with-the-wowza-streaming-cloud-rest-api) for information on query parameters.

<table>
<tr><th>v1.0</th><th>v1.3</th></tr>
<tr>
<td>A request to an index, for example,
`GET /live_streams`
returned detailed information about every live stream in the account.
Copy
<pre>
{
  "live_streams": [
    {
      "aspect_ratio_height": 1080,
      "aspect_ratio_width": 1920,
      "billing_mode": "pay_as_you_go",
      "broadcast_location": "us_west_california",
      "closed_caption_type": "cea",
      "connection_code": "0e15cb",
      ...
      "stream_targets": [],
      "target_delivery_protocol": "hls",
      "transcoder_type": "transcoded",
      "updated_at": "2017-12-27T19:27:16.316Z",
      "video_fallback": false
    },
    {
      "aspect_ratio_height": 1080,
      "aspect_ratio_width": 1920,
      "billing_mode": "twentyfour_seven",
</pre>
</td>
<td>A request to an index, for example,
GET /live_streams
returns limited information about every live stream in the account. Responses with over 1000 entries
are paginated by default.
Copy
<pre>
{
  "live_streams": [
    {
    "id": "wdjfqvsv",
    "name": "My PAYG Transcoded WSE Stream",
    "created_at": "2019-02-27T16:07:56.139Z",
    "updated_at": "2019-02-28T22:06:54.139Z"
    },
    {
    "id": "kyxwktgq",
    "name": "My 24x7 Passthrough GoCoder Stream",
    "created_at": "2019-02-27T16:07:56.139Z",
    "updated_at": "2019-03-01T09:31:08.139Z"
    },
    {
    "id": "ly40zdsg",
    "name": "My PAYG Transcoded Teradek Stream",
    "created_at": "2019-02-27T16:07:56.139Z",
    "updated_at": "2019-03-01T06:59:40.139Z"
</pre>
</td>
</tr>
</table>

v1.0

```
      "broadcast_location": "eu_germany",
      ...
      "stream_targets": [],
      "target_delivery_protocol": "hls-hds",
      "transcoder_type": "passthrough",
      "updated_at": "2017-12-28T11:01:09.316Z",
      "use_stream_source": false,
      "video_fallback": false
    },
    {
      "aspect_ratio_height": 1080,
      "aspect_ratio_width": 1920,
      "billing_mode": "pay_as_you_go",
      "broadcast_location": "us_east_virginia",
      ...
      "stream_targets": [],
      "target_delivery_protocol": "hls-https",
      "transcoder_type": "transcoded",
      "updated_at": "2017-12-28T09:56:49.316Z",
      "use_stream_source": true,
      "video_fallback": false
    },
    ...
  ]
}
```

v1.3

```
    "updated_at": "2019-03-01T06:59:49.139Z"
  },
  ...
 ]
}
```

To retrieve details about a live stream, identify the ID of the live stream and use it in a second request.

`GET /live_streams/[`*`live_stream_id`*`]`

This returns all available details of a single live stream.

Copy

```
{
  "live_stream": {
    "aspect_ratio_height": 1080,
    "aspect_ratio_width": 1920,
    "billing_mode": "pay_as_you_go",
    "broadcast_location": "us_west_california",
    "closed_caption_type": "cea",
    "connection_code": "0e15cb",
    ...
    "stream_targets": [],
    "target_delivery_protocol": "hls",
    "transcoder_type": "transcoded",
    "updated_at": "2019-01-29T07:45:46.878Z",
    "use_stream_source": true
  }
}
```

Compare workflows for each updated endpoint:

v1.0	v1.3
GET /live_streams	GET /live_streams GET /live_streams/[*live_stream_id*]
GET /transcoders	GET /transcoders GET /transcoders/[*transcoder_id*]
GET /players	GET /players GET /players/[*player_id*]
GET /recordings	GET /recordings GET /recordings/[*recordings_id*]
GET /transcoders/[*transcoder_id*]/recordings	GET /transcoders/[*transcoder_id*]/recordings GET /recordings/[*recording_id*]
GET /stream_sources	GET /stream_sources GET /stream_sources/akamai/[*stream_source_id*]
GET /stream_targets	GET /stream_targets GET /stream_targets/custom/[*target_id*] or GET /stream_targets/wowza/[*target_id*] or GET /stream_targets/ull/[*target_id*]

Stream target endpoint syntax and object references

In v1.3, we've broken out stream targets into three sets of endpoints based on the type of stream target. Custom targets, Wowza targets, and ultra low latency targets have separate endpoints with corresponding create, retrieve, update, and delete functionality. See the v1.3 Stream Targets of the API reference documentation for more information.

v1.0	v1.3
POST /stream_targets	POST /stream_targets/custom POST /stream_targets/wowza POST /stream_targets/ull
GET /stream_targets/[*target_id*]	GET /stream_targets/custom/[*target_id*] GET /stream_targets/wowza/[*target_id*] GET /stream_targets/ull/[*target_id*]
PATCH /stream_targets/[*target_id*]	PATCH /stream_targets/custom/[*target_id*] PATCH /stream_targets/wowza/[*target_id*] PATCH /stream_targets/ull/[*target_id*]
DELETE /stream_targets/[*target_id*]	DELETE /stream_targets/custom/[*target_id*] DELETE /stream_targets/wowza/[*target_id*] DELETE /stream_targets/ull/[*target_id*]

We've also updated object references for custom stream targets, Wowza stream targets, and ultra low latency stream targets. These didn't exist as separate objects in v1.0 because they were part of the stream target object.

v1.0	v1.3

Copy
`{`
`  "stream_target": {`
`    ...`
`  }`
`}`
	Copy
	`{`
	`  "stream_target_custom": {`
	`    ...`
	`  }`
	`}`
	Copy
	`{`
	`  "stream_target_wowza": {`
	`    ...`
	`  }`
	`}`
	Copy
	`{`
	`  "stream_target_ull": {`
	`    ...`
	`  }`
	`}`

Stream source endpoint syntax and object references

In v1.3, we've updated the endpoint syntax for stream sources to allow for growth with more stream source types. Although functionality remains the same, stream sources from v1.0 are now Akamai stream sources in v1.3.

> **Important**: As of January 16, 2020, Akamai stream sources are no longer functional. Akamai has withdrawn support for this feature. To prevent a disruption of service, you must update existing stream source configurations. For more information, see Remove an Akamai stream source from a live stream or transcoder using the Wowza Streaming Cloud REST API (/help/remove-an-akamai-stream-source-from-a-live-stream-or-transcoder-using-the-wowza-streaming-cloud-rest-api).

v1.0	**v1.3**
GET /stream_sources/[*stream_source_id*]	GET /stream_sources/akamai/[*stream_source_id*]
DELETE /stream_sources/[*stream_source_id*]	DELETE /stream_sources/akamai/[*stream_source_id*]

We've also updated object references for Akamai stream sources.

v1.0	**v1.3**

Copy
`{`
`  "stream_source": {`
`    ...`
`  }`
`}`
	Copy
	`{`
	`  "stream_source_akamai": {`
	`    ...`
	`  }`
	`}`

Pagination

In v1.3, queries to retrieve all live streams, players, recordings, schedules, stream sources, stream targets, and transcoders have pagination enabled by default. There are maximum of 1000 records displayed at one time. We updated default values for pagination query parameters; *page* defaults to **1**, and *per_page* defaults to **1000**. See Get paginated query results with the Wowza Streaming Cloud REST API (/help/how-to-get-paginated-query-results-with-the-wowza-streaming-cloud-rest-api) for more information.

Deprecations

In addition to deprecations related to endpoint syntax changes for stream targets and stream sources, we deprecated the following parameters and operations.

v1.0 (Deprecated)	**v1.3 (Use instead)**
POST /transcoders/[*transcoder_id*]/outputs/[*output_id*]/add_stream_target	POST /transcoders/[*transcoder_id*]/outputs/[*output_id*]/output_stream_targets
DELETE /transcoders/[*transcoder_id*]/outputs/[*output_id*]/remove_stream_target	DELETE /transcoders/[*transcoder_id*]/outputs/[*output_id*]/output_stream_targets/[*id*]
POST /stream_sources/add	POST /stream_sources/akamai
chunk_size parameter for stream targets	Use the **chunkSize** property with POST /stream_targets/[*stream_target_id*]/properties
use_https parameter for stream targets	Use the **relativePlaylists** property with POST /stream_targets/[*stream_target_id*]/properties
video_fallback parameter for transcoders and live streams	*video_fallback* was in preview, and there isn't a suggested replacement
transcoding_uptime_id parameter for transcoders, returned when starting, stopping, and fetching the state of a transcoder	Starting, stopping, or fetching a transcoder returns the *uptime_id* property instead

state of a transcoder

v1.0 (Deprecated)	**v1.3 (Use instead)**

Added and updated functionality

Check out the other updates and additions to the REST API between v1.0 and v1.3. Links go to the applicable sections of reference documentation unless otherwise noted.

- Updated *region_override* values and added the *state* parameter for ultra low latency stream targets
- Updated the *type* values listed when requesting all stream targets to **custom**, **wowza**, and **ull**
- Added the ability to filter query results when fetching all transcoders
- Added the *ip_address* parameter to the response received when fetching the state of a transcoder or live stream. See Speed up the connection to a transcoder with the Wowza Streaming Cloud REST API for details on how to use it.
- Added **fitMode** transcoder property
- Added **convertAMFData**, **redundantChunklists**, and **playlistSeconds** stream target properties
- Added REST API access to the 30-day free trial of Wowza Streaming Cloud
- Enhanced performance for API processes - for example, faster stream target creation time

More resources

- Wowza Streaming Cloud REST API release notes
- About the Wowza Streaming Cloud REST API
- Start building