Get started streaming to Wowza CDN on Fastly with the Wowza Streaming Cloud REST API

Learn how to use the Wowza Streaming Cloud™ REST API to configure a live stream or transcoder to deliver streams over Wowza CDN on Fastly. This configuration provides an alternative integrated CDN option for reliable delivery in regions around the world.

About streaming over Wowza CDN on Fastly

With Wowza CDN on Fastly, you can reliably scale your streams to reach global audiences of any size using an integration between Wowza Streaming Cloud and Fastly CDN. Because Wowza CDN on Fastly is integrated directly with Wowza Streaming Cloud, you don’t need a separate CDN provider subscription. For a feature comparison of Wowza CDN on Fastly stream targets and other supported target types, see Stream target features and properties in Wowza Streaming Cloud.

We recommend using Wowza CDN on Fastly stream targets for new stream configurations. You can also update existing live streams or transcoders by deleting all stream targets associated with outputs and replacing them with Wowza CDN on Fastly targets. Your HLS playback URLs will change during this process, but source connection information will remain the same.

**Note:** Ultra low latency streams aren’t compatible with Wowza CDN on Fastly.

Create a live stream

By default, Wowza CDN on Fastly stream targets are automatically provisioned when you create Wowza Streaming Cloud live streams.

Create a live stream using the Wowza Streaming Cloud REST API | live_streams endpoint. The new live stream uses a Wowza CDN on Fastly stream target. For
detailed instructions, see Get started broadcasting a live stream using the Wowza Streaming Cloud REST API.

Next, you can **Start and stop the stream**.

## Create a transcoder

To use a Wowza CDN on Fastly stream target with a transcoder, start by configuring the transcoder.

1. Using the Wowza Streaming Cloud REST API, create a transcoder and add output renditions. For instructions, see:
   - Create an ABR stream and send it to a target with the Wowza Streaming Cloud REST API
   - Pass a stream through the transcoder to a target with the Wowza Streaming Cloud REST API
2. Make note of the ID of the transcoder and the ID of each output rendition. You’ll need them to assign a Wowza CDN on Fastly stream target to the output renditions of the transcoder.

## Create a Wowza CDN on Fastly stream target for the outputs

Next, you’ll create the Wowza CDN on Fastly stream target, which is a destination that uses Wowza CDN to deliver the stream to players via Fastly.

### Wowza CDN on Fastly stream target parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>string</td>
<td>(Required) A descriptive name for the stream target. Maximum 255 characters.</td>
</tr>
<tr>
<td>force_ssl_playback</td>
<td>Boolean</td>
<td>If <strong>true</strong>, requires users to play the stream over HTTPS. If <strong>false</strong>, the default, users can play the stream over HTTPS or HTTP.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Data Type</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>geoblock_enabled</td>
<td>Boolean</td>
<td>If <strong>true</strong>, controls access to the stream from specific locations and, optionally, IP addresses. The default is <strong>false</strong>.</td>
</tr>
<tr>
<td>geoblock_by_location</td>
<td>string</td>
<td>Specifies whether to <strong>allow</strong> or <strong>deny</strong> access to the stream from specific locations. The default is <strong>disabled</strong>.</td>
</tr>
<tr>
<td>geoblock_country_codes</td>
<td>string</td>
<td>Required when geoblock_by_location is <strong>allow</strong> or <strong>deny</strong>. The locations affected by the geo-blocking. Enter a comma-separated list of uppercase two-letter ISO 3166-1 country codes. For a list, see ISO 3166-1 on Wikipedia.</td>
</tr>
<tr>
<td>geoblock_ip_override</td>
<td>Boolean</td>
<td>Specifies whether specific IP addresses can override the locations that are allowed or restricted. <strong>allow</strong> permits access from IP addresses within a location that's been blocked, while <strong>deny</strong> restricts access from IP addresses within locations that are allowed. The default is <strong>disabled</strong>.</td>
</tr>
<tr>
<td>geoblock_ip_addresses</td>
<td>string</td>
<td>Required when geoblock_ip_override is <strong>allow</strong> or <strong>deny</strong>. Enter addresses that can access or restrict the stream even if they're within a specified geo-blocked location. Use a comma-separated list of IPv4 IP addresses that always allow or deny streaming based on the geoblock_ip_override value.</td>
</tr>
<tr>
<td>token_auth_enabled</td>
<td>Boolean</td>
<td>If <strong>true</strong>, token authentication is enabled. If <strong>false</strong>, it's disabled. The default is <strong>false</strong>.</td>
</tr>
</tbody>
</table>

A trusted, shared secret for token authentication. It must contain only hexadecimal characters and be an even
**token_auth_shared_secret**  
*Data Type:* string  
*Description:* number of characters equal to or less than 32. If you enable token authentication but don’t set a **token_auth_shared_secret** value, the Wowza Streaming Cloud service generates the value for you.

**token_auth_playlist_only**  
*Data Type:* Boolean  
*Description:* If **true**, Wowza Streaming Cloud protects the master playlist only and leaves individual media playlists and media segments unprotected. If **false**, the master playlist, media playlists, and media segments are all protected. This feature enables playback compatibility with media players that don’t support the `withCredentials` property. It may also be useful when addressing token auth compatibility issues with specific browsers. The default is **false**.

---

**Example request and response**

Create a Wowza CDN on Fastly stream target:

```bash
curl -X POST "
-H "Content-Type: application/json"
-H "wsc-api-key: $WSC_API_KEY"
-H "wsc-access-key: $WSC_ACCESS_KEY"
-d '{
    "stream_target_fastly": {
        "name": "My first Wowza CDN on Fastly target"
    }
}' "$WSC_HOST/api/$WSC_VERSION/stream_targets/fastly"
```

The details of the configured target are listed in the response, which should look something like this:
Add the stream target to each output

After creating the Wowza CDN on Fastly stream target, associate it with each output rendition for your transcoder by creating an output stream target object.

Output stream target parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>output_id</td>
<td>string</td>
<td>The unique alphanumeric string that identifies the output rendition. You can find the ID in the details of each of the outputs you just created.</td>
</tr>
<tr>
<td>stream_target_id</td>
<td>string</td>
<td>The unique alphanumeric string that identifies the stream target that will deliver the stream to viewers. You can find the ID in the details of the custom stream target you just created.</td>
</tr>
<tr>
<td>transcoder_id</td>
<td>string</td>
<td>The unique alphanumeric string that identifies the transcoder that will generate the output renditions. You can find the ID in the details of the transcoder you just created.</td>
</tr>
</tbody>
</table>
Example request and response

Assign a stream target to an output:

```
curl -X POST \
-H "Content-Type: application/json" \
-H "wsc-api-key: $(WSC_API_KEY)" \
-H "wsc-access-key: $(WSC_ACCESS_KEY)" \
-d '{
  "output_stream_target": {
    "stream_target_id": "lwzgrj9r"
  }
}' '${WSC_HOST}/api/${WSC_VERSION}/transcoders/[transcoder_id]/outputs/[output_id]/output_stream_targets'
```

The details of the assigned target are listed in the response, which should look something like this:

```
{
  "output_stream_target": {
    "stream_target_id": "lwzgrj9r"
  }
}
```

Start and stop the stream

Now you can connect the configured live stream or transcoder to the source encoder or camera and test the setup. For more information on setting up a video source for Wowza Streaming Cloud, see Connect a source.

Use the PUT method to start and stop the live stream or transcoder. Wowza Streaming Cloud ingests the stream from the source, creates the passthrough output and the lower-quality transcoded output renditions, and sends all of the outputs to Wowza CDN via Fastly.

Example requests

Start the live stream:

```
curl -X PUT \
-H "wsc-api-key: $(WSC_API_KEY)" \
-H "wsc-access-key: $(WSC_ACCESS_KEY)" \
"${WSC_HOST}/api/${WSC_VERSION}/live_streams/[live_stream_id]/start"
```
Stop the live stream:

```bash
curl -X PUT \
-H "wsc-api-key: $WSC_API_KEY" \
-H "wsc-access-key: $WSC_ACCESS_KEY" \
"${WSC_HOST}/api/${WSC_VERSION}/live_streams/[live_stream_id]/stop"
```

Start the transcoder:

```bash
curl -X PUT \
-H "wsc-api-key: $WSC_API_KEY" \
-H "wsc-access-key: $WSC_ACCESS_KEY" \
"${WSC_HOST}/api/${WSC_VERSION}/transcoders/[transcoder_id]/start"
```

Stop the transcoder:

```bash
curl -X PUT \
-H "wsc-api-key: $WSC_API_KEY" \
-H "wsc-access-key: $WSC_ACCESS_KEY" \
"${WSC_HOST}/api/${WSC_VERSION}/transcoders/[transcoder_id]/stop"
```

Apply advanced properties

Wowza CDN on Fastly stream targets support a subset of advanced properties available for Wowza CDN on Akamai HLS stream targets and for custom stream targets whose `provider` is `akamai_cupertino`. You can use the `chunkSize`, `convertAMFData`, `playlistSeconds`, and `relativePlaylists` properties with Wowza CDN on Fastly stream targets. By default, these protocol properties are enabled and configured to use their default values. You can, however, change or customize them. For information, see Set advanced properties with the Wowza Streaming Cloud REST API.

Test playback

For a live stream, if you've enabled a hosted page, use the `hosted_page_url` in the live stream details to play a test stream. If you don’t have a hosted page enabled, use the `player_hls_playback_url` in the live stream details to configure and play a test stream using an HLS-compatible player such as Wowza™ Player.
For a transcoder, use the `playback_url` available in the Wowza CDN on Fastly stream target details to configure and play a test stream using an HLS-compatible player such as Wowza Player.

## Analyze usage data

Usage data for Wowza CDN on Fastly stream targets updates approximately in or near real-time. You can query for a summary of stream target usage, detailed usage for all stream targets, detailed usage for a single stream target, or the number of real-time viewers of an active stream.

### Wowza CDN on Fastly stream target usage query parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>from</td>
<td>The start of the range of time you want to view. Specify <code>YYYY-MM-DD HH:00:00</code> where <code>HH</code> is a 24-hour clock in UTC. The range doesn’t include hours and minutes, and it rounds minutes up to the hour. The maximum difference between from and to is 90 days. You can set the from query parameter without setting the to query parameter. The <code>from</code> default is the last billing date.</td>
</tr>
<tr>
<td>to</td>
<td>The end of the range of time you want to view. Specify <code>YYYY-MM-DD HH:00:00</code> where <code>HH</code> is a 24-hour clock in UTC. The range doesn’t include hours and minutes, and it rounds minutes up to the hour. The maximum difference between <code>from</code> and <code>to</code> is 90 days. You can set the <code>to</code> query parameter without setting the <code>from</code> query parameter. The <code>to</code> default is the end of the current day.</td>
</tr>
</tbody>
</table>

### Example requests

#### Fetch a summary of stream target usage for the account:

```bash
curl -X GET \
-H "wsc-api-key: ${WSC_API_KEY}" \
-H "wsc-access-key: ${WSC_ACCESS_KEY}" \
"${WSC_HOST}/api/${WSC_VERSION}/usage/stream_targets/summary?from=2019-08-04 15:00:00&to=2019-09-04 15:00:00"
```

The response provides aggregate account information for all Wowza CDN on Fastly
stream targets, including unique viewers, viewing time, and bytes of usage per billing zone.

Fetch usage details for all stream targets in the account:

```bash
curl -X GET -H "wsc-api-key: ${WSC_API_KEY}" -H "wsc-access-key: ${WSC_ACCESS_KEY}" "${WSC_HOST}/api/${WSC_VERSION}/usage/stream_targets?from=2019-08-04 15:00:00&to=2019-09-04 15:00:00"
```

**Note:** The response is paginated with a default `per_page` value of 1000. For more information on paginating query results, see [Get paginated query results with the Wowza Streaming Cloud REST API](#).

The response provides detailed information for all Wowza CDN on Fastly stream targets in the account, including the stream target ID, type of target, unique viewers, viewing time, total bytes of usage for the target, and bytes of usage per billing zone.

Fetch usage details for a single stream target:

```bash
curl -X GET -H "wsc-api-key: ${WSC_API_KEY}" -H "wsc-access-key: ${WSC_ACCESS_KEY}" "${WSC_HOST}/api/${WSC_VERSION}/usage/stream_targets/[stream_target_id]?from=2019-08-04 15:00:00&to=2019-09-04 15:00:00"
```

The response provides detailed information for a single Wowza CDN on Fastly stream target, including the stream target ID, type of target, unique viewers, viewing time, total bytes of usage for the target, and bytes of usage per billing zone.

Fetch usage data per country for a single stream target:

```bash
curl -X GET -H "wsc-api-key: ${WSC_API_KEY}" -H "wsc-access-key: ${WSC_ACCESS_KEY}" "${WSC_HOST}/api/${WSC_VERSION}/usage/stream_targets/[stream_target_id]/countries"
```

The response provides detailed usage information for the stream target, broken out by country. Details for each country where the stream was requested include the country code, country name, unique viewers, viewing time, total bytes of usage for the target, and bytes of usage per billing zone.
Fetch usage data per rendition for a single stream target:

```bash
curl -X GET \
-H "wsc-api-key: ${WSC_API_KEY}" \
-H "wsc-access-key: ${WSC_ACCESS_KEY}" \
"${WSC_HOST}/api/${WSC_VERSION}/usage/stream_targets/[stream_target_id]/renditions"
```

The response provides detailed usage information for the stream target, broken out by output rendition. Details for each rendition include the rendition name, unique viewers, viewing time, total bytes of usage for the target, and bytes of usage per billing zone.

Fetch live viewer data for a single stream target:

```bash
curl -X GET \
-H "wsc-api-key: ${WSC_API_KEY}" \
-H "wsc-access-key: ${WSC_ACCESS_KEY}" \
"${WSC_HOST}/api/${WSC_VERSION}/usage/stream_targets/[stream_target_id]/live"
```

The response provides the ID for a single Wowza CDN on Fastly stream target and the number of unique viewers for that target within the last 90 seconds.

**More resources**

- Manage HLS playback over SSL for Wowza CDN on Fastly with the Wowza Streaming Cloud REST API
- Protect a Wowza CDN on Fastly stream target with token authentication using the Wowza Streaming Cloud REST API
- Geo-block Wowza CDN on Fastly stream targets with the Wowza Streaming Cloud REST API