Create and view player URLs with the Wowza Streaming Cloud REST API

When you add a live stream, the Wowza Streaming Cloud™ service creates two playback URLs: one for viewing the stream in a browser on a computer and one for viewing the stream on a mobile device. By using the REST API, you can view and edit those URLs as well as create additional URLs for playing streams at other bitrates. For example, you can create player URLs for other output renditions that a transcoder is using.

Create a live stream

1. Using the Wowza Streaming Cloud REST API, create a live stream or a transcoder that receives video from a source encoder and delivers the stream to a hosted page or the transcoded output to a target. For instructions, see Get started broadcasting a live stream using the Wowza Streaming Cloud REST API.
2. Make note of the unique player_id that Wowza Streaming Cloud creates for your live stream.

Create a player URL

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bitrate</td>
<td>integer</td>
<td>The video bitrate, in kilobits per second (Kbps), of the output rendition that will be played at the URL. May correspond to the bitrate of an output rendition being used by the live stream or transcoder. Must be greater than 0.</td>
</tr>
<tr>
<td>height</td>
<td>integer</td>
<td>The height, in pixels, of the output rendition that will be played at the URL. May correspond to the height of an output rendition being used by the live stream or transcoder. Must be greater than 0.</td>
</tr>
<tr>
<td>label</td>
<td>string</td>
<td>A descriptive name for the player URL. Maximum 255 characters.</td>
</tr>
<tr>
<td>url</td>
<td>string</td>
<td>The URL of the player. If using Wowza CDN on Fastly targets, the HLS playback URL format is https://[wowzasubdomain].wowza.com/1/[stream_id]/[stream_name]/hls/live/playlist.m3u8 If using Wowza CDN on Akamai targets, the HLS playback URL format is https://[wowzasubdomain].akamaihd.net/i/[stream_name]_[angle]@[stream_id]/master.m3u8</td>
</tr>
<tr>
<td>width</td>
<td>integer</td>
<td>The width, in pixels, of the output rendition that will be played at the URL. May correspond to the width of an output rendition being used by the live stream or transcoder. Must be greater than 0.</td>
</tr>
</tbody>
</table>

Example request
The request generates a player URL for the player whose ID is `1234abcd`. The URL will play an output rendition with a bitrate of 640 Kbps and a frame size of 384 x 288 pixels. The request creates a URL whose details are listed in the response, which should look something like this:

```
{
  "url": {
    "bitrate": "640",
    "height": "288",
    "label": "My384_playerurl",
    "url": "https://[wowzasubdomain].wowza.com/1/[stream_id]/[stream_name]/hls/live/playlist.m3u8",
    "width": "384"
  }
}
```

After adding or editing a player URL, you must rebuild the player in order for the changes to take effect. Use the command:

```
curl -X DELETE
-H "wsc-api-key: ${WSC_API_KEY}"
-H "wsc-access-key: ${WSC_ACCESS_KEY}"
"${WSC_HOST}/api/${WSC_VERSION}/players/[player_id]/rebuild"
```

Related requests

View a transcoder’s outputs:

```
curl -X GET
-H "Content-Type: application/json"
-H "wsc-api-key: ${WSC_API_KEY}"
-H "wsc-access-key: ${WSC_ACCESS_KEY}"
"${WSC_HOST}/api/${WSC_VERSION}/transcoders/[transcoder_id]/outputs"
```

View the details of an output rendition:

```
curl -X GET
-H "wsc-api-key: ${WSC_API_KEY}"
-H "wsc-access-key: ${WSC_ACCESS_KEY}"
"${WSC_HOST}/api/${WSC_VERSION}/transcoders/[transcoder_id]/outputs/[output_id]"
```

View a player’s URLs:

```
curl -X GET
-H "wsc-api-key: ${WSC_API_KEY}"
-H "wsc-access-key: ${WSC_ACCESS_KEY}"
"${WSC_HOST}/api/${WSC_VERSION}/players/[player_id]/urls"
```

View the details of a player URL:

```
curl -X GET
-H "wsc-api-key: ${WSC_API_KEY}"
-H "wsc-access-key: ${WSC_ACCESS_KEY}"
"${WSC_HOST}/api/${WSC_VERSION}/players/[player_id]/urls/[player_URL_id]"
```

Update a player URL:

```
curl -X POST
-H "Content-Type: application/json"
-H "wsc-api-key: ${WSC_API_KEY}"
-H "wsc-access-key: ${WSC_ACCESS_KEY}"
-d '{
  "url": {
    "bitrate": "640",
    "height": "288",
    "label": "My384_playerurl",
    "url": "https://[wowzasubdomain].wowza.com/1/[stream_id]/[stream_name]/hls/live/playlist.m3u8",
    "width": "384"
  }
}'
"${WSC_HOST}/api/${WSC_VERSION}/players/1234abcd/urls"
```

The request generates a player URL for the player whose ID is `1234abcd`. The URL will play an output rendition with a bitrate of 640 Kbps and a frame size of 384 x 288 pixels. The request creates a URL whose details are listed in the response, which should look something like this:

```
{
  "url": {
    "bitrate": "640",
    "created_at": "2016-03-25T11:53:28.508",
    "height": "288",
    "id": "1234abcd",
    "name": "My384_playerurl",
    "updated_at": "2016-03-25T11:53:28.508",
    "url": "https://[wowzasubdomain].wowza.com/1/[stream_id]/[stream_name]/hls/live/playlist.m3u8",
    "width": "384"
  }
}
```

After adding or editing a player URL, you must rebuild the player in order for the changes to take effect. Use the command:

```
curl -X DELETE
-H "wsc-api-key: ${WSC_API_KEY}"
-H "wsc-access-key: ${WSC_ACCESS_KEY}"
"${WSC_HOST}/api/${WSC_VERSION}/players/[player_id]/rebuild"
```
Delete a player URL:

```
curl -X PATCH
-H "Content-Type: application/json"
-H "wsc-api-key: ${WSC_API_KEY}"
-H "wsc-access-key: ${WSC_ACCESS_KEY}"
-d '{
    "url": {
        "label": "480 Modified"
    }
}' "${WSC_HOST}/api/${WSC_VERSION}/players/[player_id]/urls/[player_URL_id]"
```

Delete a player URL:

```
curl -X DELETE
-H "wsc-api-key: ${WSC_API_KEY}"
-H "wsc-access-key: ${WSC_ACCESS_KEY}"
"${WSC_HOST}/api/${WSC_VERSION}/players/[player_id]/urls/[player_URL_id]"
```