Connect an RTMP encoder to Wowza Video using the Wowza Video REST API

Updated on 05/25/2022 5:37 pm PDT

The Wowza Video™ service can connect to any H.264 encoder that supports the RTMP network protocol. RTMP is a TCP-based protocol used for low-latency streaming.

> You'll need to choose which workflow you'll use, ***live stream*** or ***transcoder***, before you begin this task. See Decide between a live stream or transcoder workflow for more information about these workflows.

Before you start

You should be familiar with the following concepts:

- **API authentication methods**. We use JSON web tokens for API authentication. See Authentication for more information.
- **Environment variables**. We use environment variables for the API version and your JWT in the cURL API request examples in this topic to make it easier for you to copy, paste, and run commands in your Terminal or Command Prompt window. If you don't set environment variables for these values, you'll need to manually enter the correct values in the code samples throughout this tutorial. See Tools for testing the API for instructions.

You should complete the following tasks:

- Install the **latest firmware installed** for your encoder.

You should have access to the following items:

- The **encoder's user guide** for details about how to operate the device or software and how to specify settings such as resolution, bitrate, and frame rate.

Live stream workflow | Transcoder workflow

1. Create a live stream

Create a live stream that receives a RTMP source, generates a player, and configures a hosted page by sending a POST request to the /live_streams endpoint.

You can use the following sample request, making sure to:

- Set *encoder* to **other_rtmp**.
- Set *broadcast_location* to the region that's closest to your video source.
- Change any values unique to your broadcast, using the API reference documentation as a resource. See the **Endpoint Reference** button below.

Sample request

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${WV_JWT}" \
-d '{
  "live_stream": {
    "aspect_ratio_height": 720,
    "aspect_ratio_width": 1280,
    "billing_mode": "pay_as_you_go",
    "broadcast_location": "us_west_california",
    "delivery_method": "push",
    "encoder": "other_rtmp",
    "name": "MyLiveStream",
    "transcoder_type": "transcoded"
  }
}' "${WV_HOST}/api/${WV_VERSION}/live_streams"
```

Endpoint Reference

Sample response

The response includes:

- An ID for the live stream that you'll use in step 3.
- *source_connection_information* you'll use in the next step to configure an RTMP source encoder for the live stream.
    - *primary_server*, *host_port*, *application*, *stream_name*, *username,* and *password*

```
{
  "live_stream": {
    "id": "1234abcd",
    "name": "MyLiveStream",
    ...
    "encoder": "other_rtmp",
    ...
    "source_connection_information": {
      "primary_server": "[subdomain].entrypoint.video.wowza.com",
      "application": "app-464B8PK6",
      "host_port": 1935,
      "stream_name": "32a5814b",
      "disable_authentication": false,
      "username": "client2",
      "password": "1234abcd"
    },
    ...
  }
}
```

2. Configure your video source

Use the *source_connection_information* from the live stream response to configure your RTMP encoder. You'll need to refer to documentation for your specific encoder to determine where to input the *source_connection_information* settings, which include the stream and user credentials for authentication.

If you were configuring OBS as the encoder, you'd enter the following stream settings in OBS:

- **URL** is formatted as:
  rtmp://[*primary_server*]:[*host_port*]/[*application*]
  Where:
    - **[*primary_server*]** is the ingest location of the server

    - **[*host_port*]** is the port (by default **1935**)
    - **[*application*]** is the application name for the stream assigned by Wowza Video
- **Stream key** is the *stream_name* value, such as b01bda67.
- **Username** and **Password** are the *username* and *password* values.

Other encoders might use different names in their user interface, like **Address** instead of **URL** and **Stream** instead of **Stream key**. Make sure to refer to your encoder's documentation to determine the correct locations.

> **Tip:** This topic uses the push delivery method. If you use the pull delivery method, configure the source by determining and providing the *source_url* value when creating the live stream. The *source_url* must be an RTMP URL with a publicly accessible hostname or IP address.

3. Test the connection

Now that you have configured your source, you can test your live stream. You'll need the *[live_stream_id]* returned in step 1.

1. Start your live stream.

```
curl -X PUT \
-H "Authorization: Bearer ${WV_JWT}" \
"${WV_HOST}/api/${WV_VERSION}/live_streams/[live_stream_id]/start"
```
Endpoint Reference

2. Check the state to make sure the live stream started.

```
curl -X GET \
-H "Authorization: Bearer ${WV_JWT}" \
"${WV_HOST}/api/${WV_VERSION}/live_streams/[live_stream_id]/state"
```
Endpoint Reference

3. Start the stream in the RTMP encoder. How you start the encoder varies by device.
4. Fetch a URL to a thumbnail that you can enter into a browser and visually confirm the stream is playing.

```
curl -X GET \
-H "Authorization: Bearer ${WV_JWT}" \
"${WV_HOST}/api/${WV_VERSION}/live_streams/[live_stream_id]/thumbnail_url"
```
Endpoint Reference

5. Stop the live stream.

```
curl -X PUT \
-H "Authorization: Bearer ${WV_JWT}" \
"${WV_HOST}/api/${WV_VERSION}/live_streams/[live_stream_id]/stop"
```
Endpoint Reference

6. Stop the stream in the source camera or encoder.

Related live stream API requests

- GET /live_streams — View all live streams for an account.
- GET /live_streams/[ID] — View the details of a live stream, including the player embed code and hosted page URL.
- PATCH /live_streams/[ID] — Update a live stream's configuration.