Connect an IP camera to Wowza Video using the Wowza Video REST API

Updated on 05/25/2022 5:40 pm PDT

IP cameras, sometimes called webcams or netcams, let you capture and broadcast live video over the internet. They're often used for video surveillance and security.

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** 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 RTSP 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 the type of camera. Valid values include:
  - **ipcamera** for a generic IP camera
  - **axis** for an Axis network IP camera
  - **sony** for a Sony SRG-300SE IP camera
- Set *source_url* according to your camera type.
  - For a generic camera, see your IP camera's user guide for for details about the *source_url*.
  - For an Axis network IP camera, the *source_url* is in the form

rtsp://[*username*]:[*password*]@[*ip_address*]/axis-media/media.amp. See your Axis camera user guide or the Axis online helpdesk for details about the camera's IP address.
    - For a Sony SRG-300SE IP camera, the *source_url* is in the form **rtsp://[camera-ip-address]/video[port]**. See the Sony SRG-300SE User's Guide for details about the camera's IP address.

      The hostname or IP address used in the *source_url* must be publicly accessible. If authentication information, such as username and password, is included in the *source_url*, it can only contain alphanumeric, period (.), underscore (_), and hyphen (-) characters.
- 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 **API endpoint** link below.

Sample request

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${WV_JWT}" \
-d '{
  "live_stream": {
    "aspect_ratio_height": 720,
    "aspect_ratio_width": 1280,
    "billing_mode": "pay_as_you_go",
    "broadcast_location": "us_west_california",
    "delivery_method": "pull",
    "encoder": "ipcamera",
    "name": "MyIPCameraLiveStream",
    "source_url": "rtsp://camera-ip-address/videoinfo",
    "transcoder_type": "transcoded"
  }
}' "${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*. Because you're pulling video from the camera, you won't need to do anything futher with this information.

```
{
  "live_stream": {
    "id": "1234abcd",
    "name": "MyIPCameraLiveStream",
    ...
    "encoder": "ipcamera",
    ...
    "source_connection_information": {
      "source_url": "rtsp://camera-ip-address/videoinfo",
    },
    ...
  }
}
```

2. Configure your video source

You provided the *source_url* that gives Wowza Video with a connection to the camera when you created the live stream. Now, make sure the IP camera's keyframe interval is set to 60 fps for NTSC or 50 fps for PAL.

3. Test your 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 RTSP 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.