Manage a broadcast from end to end with Wowza ClearCaster GraphQL API

Originally Published on 03/25/2019 | Updated on 12/13/2019 12:52 pm PST

The Wowza ClearCaster™ GraphQL API enables you to manage your Wowza ClearCaster appliance and broadcasts directly via the API. GraphQL APIs are structured so that the operations mirror the responses. We recommend using the GraphiQL app, an interactive, browser-based IDE for exploring and using GraphQL APIs.

This article describes how to create, publish, and monitor a broadcast with the Wowza ClearCaster GraphQL API.

Note: Wowza ClearCaster version 2.0.0.07 or later is required.

Identify your environment IDs

Before you can create a broadcast, you must know your namespace and device IDs. Use the following queries to obtain this and other information about your environment.
Create a broadcast

Use the createBroadcast mutation to create and configure a broadcast with a single operation. The following example creates a 720p30 broadcast and returns the new broadcast's ID.

```graphql
mutation createBroadcastBase720p30
{
  createBroadcast(name: "Demo 720p30 Broadcast",
    namespaceId: "yourNamespaceID",
    input: {
      name: "Demo 720p30 Broadcast",
      inputs: [{
        inputType: CAPTURE_HARDWARE,
        videoInput: 0,
        videoFrameWidthMax: 1280,
        videoFrameHeightMax: 720,
        videoFrameRateMax: 30,
        videoKeyFrameIntervalMilliseconds: 2000,
        overlayUrl: "overlayURL",
        broadcastInputEncoderIndex: 0
      }
    ]
    outputs: [
      }
    }
  }
}
```
streamName: "rendition0",
encodingConfiguration:
{
  name: "rendition0",
  encodingConfigurationVideo:
  {
    codec: "H.264",
    implementation: "x264",
    frameSizeFitMode: "stretch",
    frameSizeWidth: 1280,
    frameSizeHeight: 720,
    profile: "high",
    bitrate: 3500000,
    bitrateMin: 750000,
    autoAdjustBitrate: true,
    keyFrameIntervalFollowSource: true,
    parameters: [
      {
        name: "x264.preset",
        value: "4",
        type: "Long"
      },
      {
        name: "x264.ref",
        value: "1",
        type: "Long"
      },
      {
        name: "x264.bframes",
        value: "1",
        type: "Long"
      }
    ]
  },
  encodingConfigurationAudio:
  {
    codec: "AAC",
    bitrate: 96000
  }
}
streamTargets: {
  url: "[stream target URL]",
  protocol: RTMP,
  streamTargetEncoderIndex: 0
}
}
}
broadcastEncoders: [{
  encoderId: "[encoder ID]",
  streamTargetEncoderIndex: 0
}
Manage the broadcast

You can use the following query to look up information about your broadcast at any time.

```graphql
query broadcasts {
  broadcast(id: "yourBroadcastId") {
    id
    name
    createdAt
    updatedAt
    previewedAt
    liveAt
    stoppedAt
    inputs {
      inputType
      videoInput
      videoFrameWidthMax
      videoFrameHeightMax
      videoFrameRateMax
      videoAspectRatioMode
      videoAspectRatioWidth
      videoAspectRatioHeight
      videoAspectRatioRotation
      audioLevel
      overlayVendor
      overlayUrl
    }
    broadcastEncoders {
      id
      streamTargetEncoderIndex
      broadcastInputEncoderIndex
      encoder {
        id
        deviceId
        name
      }
    }
  }
}
```
Publish the broadcast

Before you can publish a broadcast, you must activate an encoder to subscribe to the broadcast. In doing so, the encoder receives all of the encoding configuration settings specified for the broadcast. Then you can publish the broadcast by setting the broadcast status to live.

Monitor the broadcast

The Wowza ClearCaster GraphQL API provides a variety of near real-time stream health metrics to help you ensure that your broadcasts run smoothly. The metrics are also available after the broadcast ends. For more information, see Monitor Wowza ClearCaster broadcasts with the GraphQL API.

End the broadcast
After your broadcast ends, use the setBroadcastStatus and setDeactivateBroadcastEncoders mutations to end the broadcast.