# Deliver real-time streams with the Wowza Video REST API

Updated on 05/30/2023 11:22 am PDT

Wowza Video™ Real-Time Streaming at Scale provides half-second latency to all your viewers, no matter where they are. Real-time streaming is perfect for interactive use cases like video chats, auctions, e-sports, fitness, e-commerce, gambling, and more.

If your audience is fewer than 300 viewers or you want to stream in near real-time alongside other delivery protocols, you may choose to use our standard WebRTC solution.

**Note**: Real-Time Streaming at Scale is available only through the 1.7 version of the Wowza Video REST API or later.

Streaming using Real-Time Streaming at Scale requires that you provision the stream. The stream can be supplied from Wowza Video using the UI or API. See Deliver real-time streams to viewers with Wowza Video for instructions on how to use the Wowza Video UI to create a Real-Time stream.

We have three options for configuring your streaming publish and playback clients: the Javascript SDK for Real-Time Streaming, OBS fo Real-Time Streaming at Scale, and RTMP. You need to create and host a viewer page using HMTL and Javascript to use Real-Time Streaming at Scale with any configuration.

You must use the Javascript SDK to stream playback. Alternatively, consider using the Wowza Flowplayer Real-Time Streaming plugin for playback if you need to embed and configure the player on your own page. Wowza Flowplayer is included as part of your Wowza Video subscription.

## 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 (${WV_VERSION}) and your JWT (${WV_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.
- **Real-time streaming workflows**. See About real-time streaming for more information.
- **HTML and Javascript**.

You should have access to the following items:

- A **RTS@S license**. Contact us for more information. To enable and purchase capacity for Real-Time Streaming at Scale for your account and access the /real_time operations, contact 720.279.8163 or schedule a call.

## 1. Create a real-time stream

Create a real-time stream by sending a POST request to the /real_time endpoint.

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

- Set name to a descriptive name for your real-time stream.

  > **Note**: If you save a real-time stream to Asset Management, Wowza Video uses the stream name for the recording and VOD file name. Because the stream name is used in the file name, we'll replace special characters to avoid file management issues in storage. Keep this in mind if you're looking for your asset or file on the **Recordings**, **VOD Stream**, and **Manage Assets** list pages and you see some differences in what you entered vs what's displayed.

- Set enable_secure_viewer to true and set an expires_on date if you would like to add a security token that must be passed by viewers for playback.
- Change any values unique to your broadcast, using the API reference documentation as a resource. See the **Endpoint Reference** button below for a full list of options. Some examples:
  - You might set recording to true so you'll have an MP4 of the stream you can download later. See Record a real-time stream with the Wowza Video REST API for more information.
  - If your broadcast location is closest to Europe, Middle East, or Africa, set region to amsterdam for the best stream performance possible. The default is phoenix.
  - If your broadcast location is closest to East Asia, South Asia, Southeast Asia, or Oceania, set region to singapore or banaglore. The default is phoenix.
- See the Javascript SDK reference for additional configuration options.

### Sample request

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${WV_JWT}" \
-d '{
  "real_time_stream": {
    "name": "MyRealTimeStream",
    "enable_secure_viewer": false
  }
}' "${WV_HOST}/api/${WV_VERSION}/real_time"
```

Endpoint Reference

### Sample response

The response includes:

- An ID that you can use to perform other actions on the stream, like getting the details or deleting it. This is also used to configure the viewer page.
- The stream_name generated by Wowza Video. This is used to configure the viewer page.

```
{
 "real_time_stream": {
  "id": "2adffc17",
  "name": "MyRealTimeStream",
  "stream_name": "8d304b93f1684320a54f2798666eeca7",
  "token": "97e52731bc21ef66e4c05a8ee1e28b64bf5f9db728573d94e690277cea9215bc",
  "subscribe_token": "50e161bfa42fbd581a0dfe5f632596b86c2c577a56bd439b38f8c904aabad04d",
  "rtmp_url": "rtmp://[primary_server]:[host_port]/[sub-domain]/[stream_name]?[token]",
  "enable_secure_viewer": true,
  "state": "active",
  "recording": false,
  "disable_vod_encoder": false,
  "created_at": "2021-06-30T18:02:20.00Z",
  "updated_at": "2021-06-30T20:03:16.00Z",
  "region": "phoenix"
 }
}
```

Wowza Video creates a real-time stream and provides a **token** property for publishing security, a **subscribe_token** property for viewing security, and **stream_name** you will need to configure your source.

## 2. Configure your video source

Now that you have created your stream, you have three options for configuring input to your real-time stream:

- **Javascript SDK for Real-Time streaming** – Embeds your browser-based capture into a web page.
- **OBS for Real-Time Streaming at Scale – Wowza WebRTC** – Gives you the flexibility to use multiple inputs, transitions, and other production tools traditionally provided by OBS.
- **RTMP** – Supports all professional devices. This protocol is not optimized for low latency, so it is likely to increase the streaming delay by approximately 1 second.

### JavaScript SDK for Real-Time Streaming

The Javascript SDK lets you build your own custom publication page. We provide sample code to get you started.

> You can also use NPM to build your publication and viewer pages. See the Javascript SDK reference for more information.

Use the publisher page sample code, making sure to:

- Set 'my-stream-name' to the stream_name generated by Wowza for your real-time stream, for example: *8d304b93f1684320a54f2798666eeca7*
- Set 'my-publish-token' to the token generated by Wowza for your real-time stream, for example: *97e52731bc21ef66e4c05a8ee1e28b64bf5f9db728573d94e690277cea9215bc*
- See the Javascript SDK reference for additional configuration options.

```
<html>
 <head>
  <script src='https://www.wowza.com/downloads/rts-sdk/wowza.umd.js'></script>
 </head>
 <body>
  <div>
    <h1>Publish</h1>

    <video autoplay playsinline controls muted id="my-video" width="1280" height="720"></video>

    <script>
      const wowza = window.wowza
      const streamID = 'my-stream-name'
      const tokenID = 'my-publish-token'

      //Define callback for generate new tokens
      const tokenGenerator = () => wowza.Director.getPublisher({
        token: tokenID,
        streamName: streamID
       })

      //Create a new instance
      const wowzaPublish = new wowza.Publish(streamID, tokenGenerator)

      //Get User camera and microphone
      navigator.mediaDevices.getUserMedia({ audio: true, video: true }).then(
        (mediaStream) => {
          //Publishing Options
          const broadcastOptions = {
            mediaStream
          }

          document.getElementById('my-video').srcObject = mediaStream

          //Start broadcast
          try {
            wowzaPublish.connect(broadcastOptions)
          } catch (e) {
            console.log('Connection failed, handle error', e)
          }
        }
      )

    </script>
  </div>
 </body>
</html>
```

OBS – Wowza WebRTC

1. Install OBS for Real-Time Streaming at Scale for your operating system.

   **Debian/Ubuntu:**
   https://www.wowza.com/downloads/wowza-obs/wowza-obs-1.4.2-28.1.2-m104-Linux-2004.deb
   **Debian/Ubuntu:**
   https://www.wowza.com/downloads/wowza-obs/wowza-obs-1.4.2-28.1.2-m104-Linux-2204.deb

   **Note**: If you see the following error when you install, the installation likely still succeeded: `N: Download is performed unsandboxed as root as file '/home/user/wowza-obs-1.4.2-28.1.2-m104-Linux-2204.deb'` `couldn't be accessed by user '_apt'. - pkgAcquire::Run (13: Permission denied)`

**Windows:**
https://www.wowza.com/downloads/wowza-obs/wowza-obs-x64-1.4.2-28.1.2-m104.msi
**Mac:**
https://www.wowza.com/downloads/wowza-obs/Wowza-OBS-Real-Time-1.4.2-28.1.2-m104-macos-arm64.dmg
**Mac:**
https://www.wowza.com/downloads/wowza-obs/Wowza-OBS-Real-Time-1.4.2-28.1.2-m104-macos-x86_64.dmg

2. Click **Settings**, click **Stream**, and select **Wowza WebRTC** as the **Service**.
3. Enter the **Stream Name** and **Publishing Token** provided by Wowza Video, and then click **OK**.

RTMP

Use the *rtmp_url* 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 *rtmp_url* settings.

> Be sure you have the latest firmware installed for your encoder.

Common stream settings for encoders include:

- ***[primary_server]*** is the ingest location of the server, such as rtmp://rtmp-realtime1.wowza.com:1935/v2/pub/
- ***[host_port]*** is the port (by default **1935**)
- **Stream key** is the *stream_name* value, such as `6c6f5e28cfaa4467b16b47716199847c?token=70gee2f5f208615428acb633c2485d8595c90f98e677e11c55ae2c7fd71ecea2`

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 stream. The *source_url* must be an RTMP URL with a publicly accessible hostname or IP address.

## 3. Configure your viewer page

The Javascript SDK lets you build your own custom viewer page. We provide sample code below to get you started.

> **Tip:** Consider using the Wowza Flowplayer Real-Time Streaming (WebRTC) plugin for playback instead of the Javascript SDK. The code samples are included on the plugin page. Wowza Flowplayer is included as part of your Wowza Video subscription.

Use this viewer page sample code, making sure to:

- Set 'my-stream-name' to the *stream_name* for the stream, for example:

*11ad1f62a39a4287895d13ac28c0a820*

```
<html>
 <head>
     <script src='https://www.wowza.com/downloads/rts-sdk/wowza.umd.js'></script>
 </head>
 <body>
   <video controls autoplay id="my-video"></video>

   <script>
     const wowza = window.wowza

     // Get Media Element
     const video = document.getElementById('my-video')

     const streamName = "my-stream-name"

     //Define callback for generate new token
     const tokenGenerator = () => wowza.Director.getSubscriber({
        streamName: streamName, subscriberToken: tokenId
     })

     //Create a new instance
     const wowzaView = new wowza.View(streamName, tokenGenerator, video)

     //Start connection to publisher
     try {
        wowzaView.connect()
     } catch (e) {

        console.log('Connection failed, handle error', e)
     }
   </script>
 </body>
</html>
```

## 4. Test the connection

1. Start your video source from the publisher page, OBS, or the encoder.
2. Open the file that contains the viewer page code in a web browser to confirm the stream is running.

## Related API requests

- GET /real_time — View all real-time streams for an account.
- GET /real_time/[ID] — View the details of a real-time stream.
- PATCH /real_time/[ID] — Update a real-time stream.
- DELETE /real_time/[ID] — Delete a real-time stream.

## More resources

- Record a real-time stream with the Wowza Video REST API