

- Overview
- Access
- Create a Stream
- Passthrough  
Encoding  
Recommendations
- Transcoding  
Settings
- Create a Stream  
(API)
- Pushing / Viewing a  
Test Stream
- Latency Testing
- Next Steps

# How to Set Up Sub-Second Streaming in Minutes



Free Trial



[docs.ceeblue.net](https://docs.ceeblue.net)



- Overview
- Access
- Create a Stream
- Passthrough  
Encoding  
Recommendations
- Transcoding  
Settings
- Create a Stream  
(API)
- Pushing / Viewing a  
Test Stream
- Latency Testing
- Next Steps

## Overview

---

This guide will provide you with step-by-step instructions on **how to set up a test stream** using Ceeblue.

The Ceeblue Media Fabric is a multiprotocol transcoding and end-to-end delivery service that supports sub-500ms latencies.

If you have any questions during setup, please contact us at [contact@ceeblue.net](mailto:contact@ceeblue.net).



- Overview
- Access**
- Create a Stream
- Passthrough  
Encoding  
Recommendations
- Transcoding  
Settings
- Create a Stream  
(API)
- Pushing / Viewing a  
Test Stream
- Latency Testing
- Next Steps

## Access

---

**Option 1:** Log in to the **dashboard** at [dashboard.ceeblue.tv/](https://dashboard.ceeblue.tv/).



Open the Dashboard

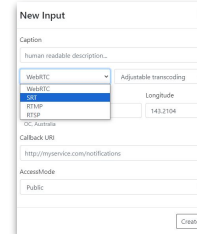
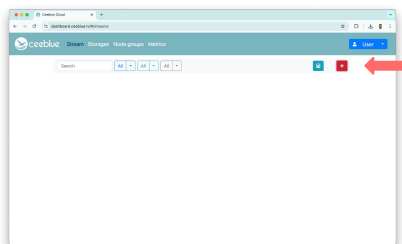
**Option 2:** To access our services via **API**, follow the instructions on this link to retrieve your JSON Web token:



How To:  
Authorization

## Create a Stream (Dashboard)

1. Click on the **Stream** tab in the menu bar. This page shows existing streams and their state.
2. Click on the “+”.
3. In the pop-up **New Input** window, select the desired **Input Protocol**.
4. Select **Adjustable Transcoding** or **Passthrough**.
5. Click on **Create**.



- Overview
- Access
- Create a Stream**
- Passthrough  
Encoding  
Recommendations
- Transcoding  
Settings
- Create a Stream  
(API)
- Pushing / Viewing a  
Test Stream
- Latency Testing
- Next Steps

## Passthrough Encoding Recommendations

 Only Applies to Passthrough Streams

### Input Protocol Considerations (WebRTC, RTMP, SRT)

- Use H.264 (baseline or constrained baseline) for WebRTC to maintain compatibility.

### H.264 Settings:

- Use “zerolateness” tuning to minimize buffering.
- Select a fast preset (e.g., “superfast”) for quick encoding. There might be a slight quality trade off.
- Set a **small GOP** (1-2 seconds) to ensure fast stream startup and keyframe delivery.
- Disable **B-frames** when targeting WebRTC, as they add delay and aren’t supported.

### Output Protocol Considerations (WebRTC, DASH, HLS)

- H.264 is the most widely supported video codec across all these protocols.

### Audio Considerations

- AAC is generally supported by DASH and HLS.
- **Opus** is commonly used in WebRTC browsers, though it’s not universal across segmented protocols.
- Use **AAC** as the common denominator if maximum compatibility is required, but use Opus if targeting WebRTC for its superior low-latency performance.

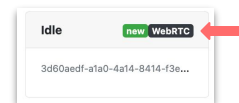
- Overview
- Access
- Create a Stream
- Passthrough Encoding Recommendations
- Transcoding Settings
- Create a Stream (API)
- Pushing / Viewing a Test Stream
- Latency Testing
- Next Steps




## Transcoding Settings (Dashboard)



 Only Applies to Transcoded Streams

1. If you chose “**Adjustable Transcoding**,” select the **new** stream by clicking anywhere on the grey bar above the Stream ID.



2. In the stream pane that opens, select the gear drop-down  and choose

 Transcoding

3. In the transcoding pane, there is a preconfigured quality ladder. These renditions can be edited or deleted. To add more quality renditions   codecs, use either the **presets** or generate **custom tracks** using the “+” button.

4. Save your changes using the green disk icon at the top of the pane. 



How To:  
Create a New Stream

Overview

Access

Create a Stream

Passthrough  
Encoding  
Recommendations

Transcoding  
Settings

Create a Stream  
(API)

Pushing / Viewing a  
Test Stream

Latency Testing

Next Steps



Clickable Menu

- Overview
- Access
- Create a Stream
- Passthrough  
Encoding  
Recommendations
- Transcoding  
Settings
- Create a Stream  
(API)**
- Pushing / Viewing a  
Test Stream
- Latency Testing
- Next Steps

## Create a Stream (API)

### Using APIs

1. Request the full list of streams.

```
curl --request GET --url https://api.ceeblue.tv/v1/inputs --header 'authorization: Bearer <your token>'
```

2. Create a new stream (with the default transcoding parameters) :

```
curl --request POST --url https://api.ceeblue.tv/v1/inputs --header 'accept: application/json' --header 'content-type: application/json' --header 'authorization: Bearer <your token>' --data '{\"format\":\"RTMP\",\"output\":{\"version\":\"2\",\"passthrough\":false,\"tracks\":[{\"type\":\"Video\",\"width\":1280,\"height\":720,\"framerate\":30,\"settings\":{\"codec\":\"H264\",\"speedPreset\":\"superfast\",\"bitrate\":2000,\"keyIntMax\":60}},{\"type\":\"Video\",\"width\":854,\"height\":480,\"framerate\":30,\"settings\":{\"codec\":\"H264\",\"speedPreset\":\"superfast\",\"bitrate\":1200,\"keyIntMax\":60}},{\"type\":\"Video\",\"width\":1280,\"height\":720,\"framerate\":30,\"settings\":{\"codec\":\"VP8\",\"bitrate\":2000,\"keyMaxDist\":60}},{\"type\":\"Video\",\"width\":854,\"height\":480,\"framerate\":30,\"settings\":{\"codec\":\"VP8\",\"bitrate\":1200,\"keyMaxDist\":60}},{\"type\":\"Audio\",\"channels\":2,\"rate\":48000,\"settings\":{\"codec\":\"AAC\",\"bitrate\":128}},{\"type\":\"Audio\",\"channels\":2,\"rate\":48000,\"settings\":{\"codec\":\"Opus\",\"bitrate\":128}}],\"access\":\"Public\"}'
```





How To:  
Create a New Stream



## Pushing / Viewing a Test Stream

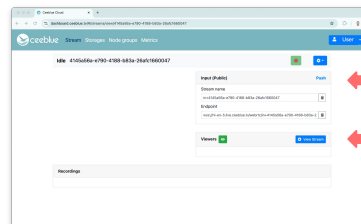
- Overview
- Access
- Create a Stream
- Passthrough Encoding Recommendations
- Transcoding Settings
- Create a Stream (API)
- Pushing / Viewing a Test Stream**
- Latency Testing
- Next Steps

**Push** When ingesting **WebRTC**, you can push using the browser by selecting “**Push**” in the Input pane.

If using another ingest protocol besides WebRTC (**RTMP, SRT, etc.**), or if you’re using WebRTC but don’t want to push from the browser, select the gear drop-down  and provision a  **GSLB Ingress Endpoint**.

You can now stream to this endpoint using OBS, vMix, etc.

**View** To view the stream output, select “**View Stream.**”




 **Push** (Opens Separate Streamer Page)

 **View Stream** (Opens Separate Viewer Page)



## Latency Testing

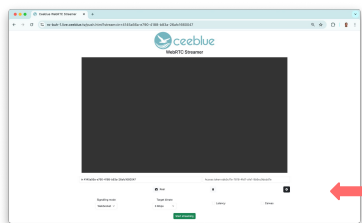
To measure end-to-end latency, select the black gear  on the **streamer page** and click on the “**Latency**” checkbox . Both “Latency” and “Canvas” will appear checked.

On the viewer page, click on the stopwatch icon . The latency measurement will appear right below the video.

### Tips:

Please note that the black and white elements at the top of the video are the mechanism by which latency is measured.

Be sure to keep both the streamer and viewer pages in the foreground to maintain performance.



Streamer Page

1. 
2. Latency



Viewer Page

- Overview
- Access
- Create a Stream
- Passthrough Encoding Recommendations
- Transcoding Settings
- Create a Stream (API)
- Pushing / Viewing a Test Stream
- Latency Testing**
- Next Steps

- Overview
- Access
- Create a Stream
- Passthrough  
Encoding  
Recommendations
- Transcoding  
Settings
- Create a Stream  
(API)
- Pushing / Viewing a  
Test Stream
- Latency Testing
- Next Steps**

## Next steps:



Set Up a Player



Start Integrating

## Let's get you started!



Contact Us!



Ceeblue Resources

