API Tutorial

Tutorial: Live stream Daily video calls with Amazon IVS for ultra-low latency

Learn how to start live streaming with the Daily API and consume the stream via Amazon IVS's player SDK



Jess Mitchell
Published: 6 Aug 2021 (Updated: 5 Jul 2022) • 5 min read



Over the past year, we've heard a higher demand for reliable video live streaming services that actually feel, well, live! At Daily, we've encountered this demand from customers hosting events like live classes, concerts, and large business meetings. In cases like these, customers are often also looking for the lowest latency possible for an optimal viewing experience.

To help accommodate this growing demand, Daily now offers live streaming for Daily calls via RTMP streaming for all paid accounts. This format can be consumed by several existing solutions (like Amazon IVS, Mux, and more) and provides low latency, ranging from 3 to 20 seconds, depending on your streaming platform.

In addition to ease of use and low latency, Daily live streaming also helps customers customize their live streaming experience. We offer several video layouts and full HD resolution. For any accounts with live streaming available, we can accommodate up to nine video feeds at a time and a configurable number of simultaneous streams per account.

To show how excited we are and how simple it really is to get live streaming up and running for any Daily call, we'll be walking through this process step-by-step. We'll use Amazon IVS in this tutorial, which we've found delivers an impressive 2-5 second latency.

By the end of this demo, you'll have the Daily prebuilt UI embedded in an app and its video playing back with Amazon IVS's player SDK.

For a primer on what live streaming is and an overview of some common terms you'll hear (like RTMP) check out our previous post on live streaming with Daily.

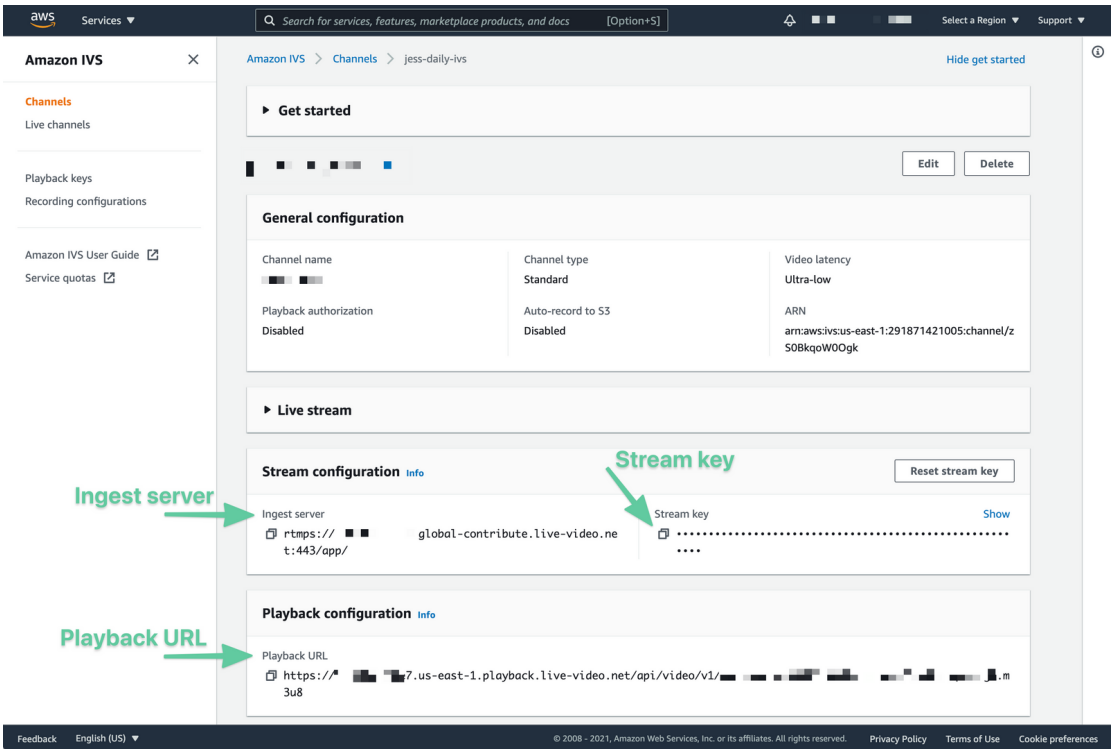
Before we dive in

To get started, there are a few requirements you will need to set up:

1. A Daily account with a credit card on file. (Live streaming is a pay-as-you-go feature.)
2. A Daily room, which can be created via the Daily dashboard or our REST API.
3. A meeting token for the Daily room, with the property `is_owner` set to `true`.
4. An AWS account.
5. Once you have an account, you can create a new streaming channel through the Amazon IVS console.

The IVS channel console will provide three pieces of important information that will be used later on. Keep the following handy for an upcoming step:

1. The ingest server
2. The stream key
3. The playback URL



Amazon IVS's streaming channel console

Embedding a Daily call with Daily's Prebuilt

Now that we have our streaming channel set up, we need to get our Daily call running. To see how this works, let's look at Daily's prebuilt demo app, which uses our prebuilt video UI.

Note: Make sure to use the `live-streaming-demo` branch in the prebuilt-ui demo.

There are two main files to be aware of in the demo:

1. index.html
2. index.js

In `index.html`, there is a script tag to include `daily-js`, Daily's front-end JavaScript library. There is also a `<div>` element that can be selected in the `index.js` file via its ID. This is where the video iframe will be inserted by `daily-js`. Since we're using Daily's prebuilt UI, that's really all we need in terms of the HTML.

In the `index.js`, there are three main steps to get to live streaming:

1. Initialize the Daily iframe with the `createFrame()` method.
2. Join the Daily room created for live streaming with the `join()` method.
3. Start live streaming by clicking the "Start live streaming" button to trigger the `startLiveStreaming()` method.

In the first step, the iframe is initialized with just a few lines of code.

```
async function setup() {
 callFrame = await window.DailyIframe.createFrame(
   document.getElementById('callframe'),
   {
     iframeStyle: {
       // ... add custom styles
     },
   }
 );
}
```

Once the Daily call has been initialized, the call can be joined (step two) by passing the room URL and meeting owner token created earlier.

Note: In the demo source code, you may notice the room URL is taken from an input. In the code example below, it's displayed as a variable to simplify things.

```
async function joinCall() {
```

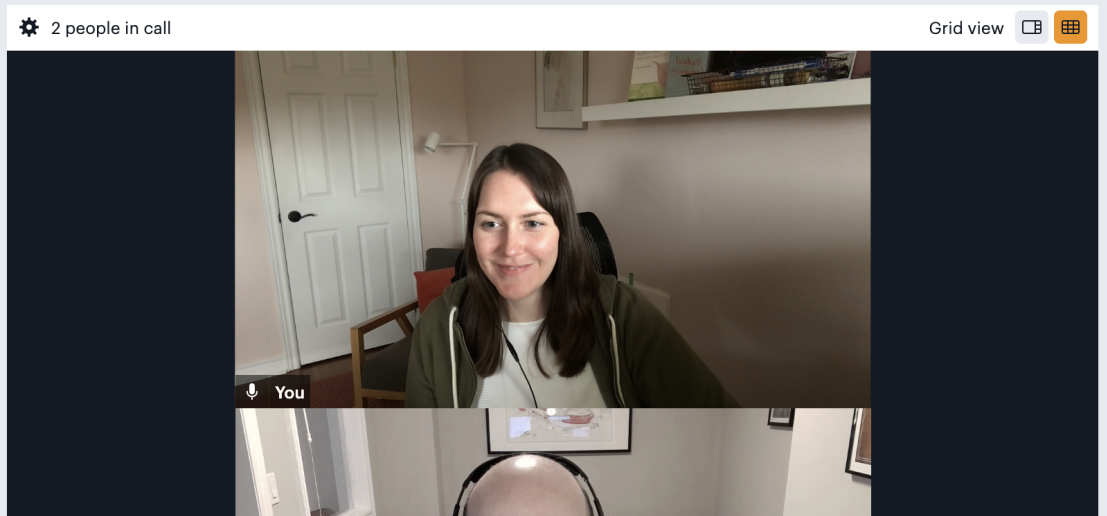
```
  const roomURL = 'DAILY_ROOM_URL';
  const token = 'MEETING_OWNER_TOKEN';
  await callFrame.join({
    url: roomURL,
    token,
  });
}
```

Once the room has been joined successfully, live streaming can be started by clicking the live streaming button in the side panel. This will call Daily's `startLiveStreaming()` method, with the RTMP URL passed as an option.

```
function startLiveStreaming() {
  callFrame.startLiveStreaming({ rtmpUrl: 'INSERT_RTMP_URL' });
// rtmpUrl = rtmps://RTMP_ENDPOINT/STREAM_KEY
}
```

The RTMP URL will use two pieces of information provided by Amazon IVS in an earlier step:

1. The ingest server (or RTMP endpoint).
2. The stream key.





The Daily Prebuilt's embedded iframe with two call participants.

Live streaming your Daily calls to the world

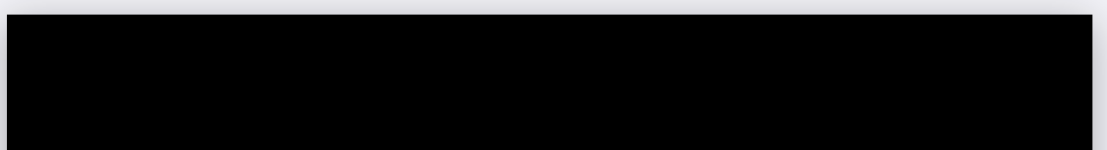
Now that you have your Daily video streaming, playing the live stream is as simple as passing the playback URL to the Amazon IVS player. Amazon offers a number of ways to quickly test the IVS player, such as the Amazon IVS basic web sample from the code repository list, or you can simply follow the player instructions for web. The latter can work with just a few lines of HTML code.

Once your player is set up, load the playback URL to the IVS player. This playback URL is one of the pieces of information provided in the stream channel's console mentioned above.

```
const playbackUrl = 'PLAYBACK_URL';


if (IVSPlayer.isPlayerSupported) {
 const player = IVSPlayer.create();
 player.attachHTMLVideoElement(document.getElementById('video-pl
 player.load(playbackUrl);
 player.play();
}
```

To confirm the live stream is available, load the player in the browser and you should see your live stream with the incredibly low latency of only 2 to 5 seconds. It's so fast we're honestly impressed ourselves!





A Daily live stream viewed through the Amazon IVS player.

Being able to live stream with such a small delay for your viewers is a game changer for live video content, especially in larger calls that otherwise would have to be restricted in size. It means helping more people connect on a larger scale and with fewer barriers for getting started. We're excited to offer this feature at Daily and can't wait to hear all the amazing ways you use it.

If you'd like to learn more about Daily live streaming or your use case requires more than 20 concurrent streams, contact us for more information.

Get updates from Daily

Don't miss a thing! Subscribe to our monthly newsletter and we'll keep you up-to-date with the latest product releases, tutorials, and more from Daily.

Email*

Country*

Join peerConnection - our video/WebRTC developer community.

By clicking submit below, you consent to allow Daily.co to store and process the personal information submitted above to provide you the content requested.

Submit

More in **API Tutorial**

Implementing cursor sharing with Daily's video call API

24 Oct 2022 - 11 min read

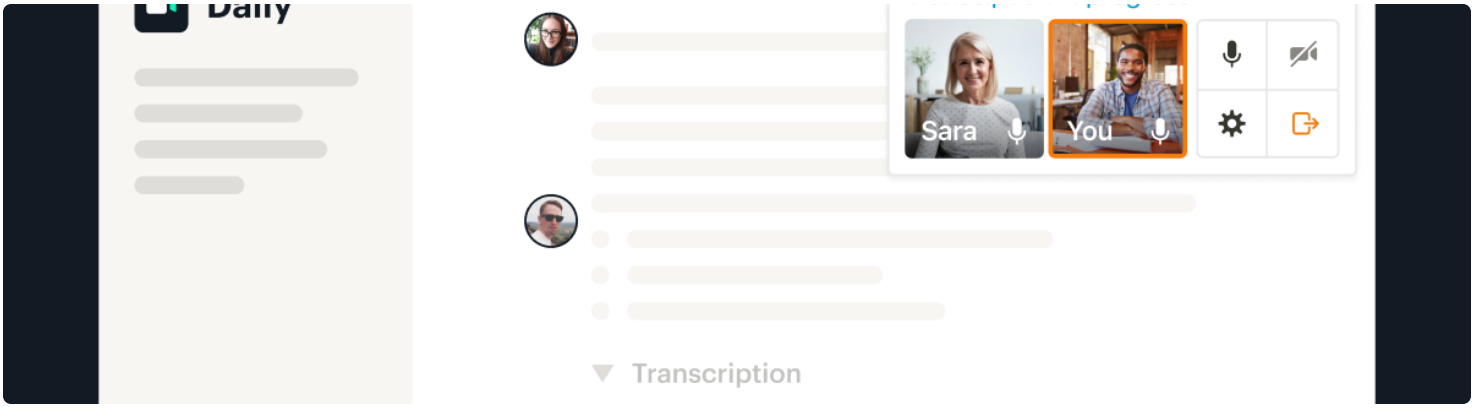
Add hand-raising to your custom video app with the Daily user data API

12 Sep 2022 - 12 min read

Stream with HLS while recording to a custom S3 bucket using Daily

30 Aug 2022 - 5 min read

See all 82 posts →



Collab Chrome Extension

Embed and transcribe a Daily WebRTC video call in any Notion page

Introducing Daily Collab, a Chrome extension demo using Daily's WebRTC video chat APIs, Notion's beta API, and Deepgram transcription



Jess Mitchell
13 Aug 2021 – 5 min read



Introducing color theming for embeddable video calls: a new feature for Daily Prebuilt

It's now even easier to integrate video calling that represents your product and brand.



Francine Boulanger
3 Aug 2021 – 2 min read