Capture a MediaStream From a Canvas, Video or Audio Element

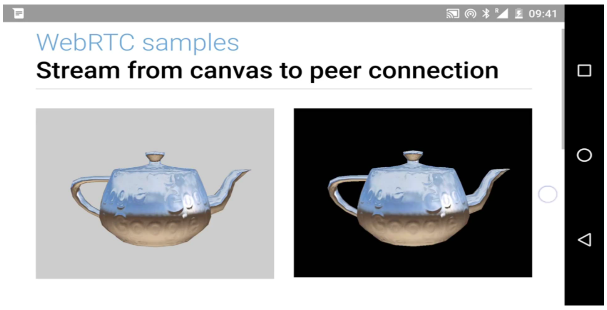
By Sam Dutton

Sam is a Developer Advocate

The `captureStream()` method makes it possible to capture a `MediaStream` from a `<canvas>`, `<audio>` or `<video>` element.

This enables a video or audio stream from any of these elements to be recorded, live-streamed via WebRTC, or combined with effects or other `MediaStreams` in a `<canvas>`. In other words, `captureStream()` enables `MediaStream` to pass media back and forth between canvas, audio or video elements — or to an `RTCPeerConnection` or `MediaRecorder`.

In the following demo (available from the WebRTC samples) a `MediaStream` captured from a canvas element on the left is streamed via a WebRTC peer connection to the video element on the right:



WebRTC samples

Stream from canvas to peer connection

(There are links to more canvas and video examples below.)

The `captureStream()` code is simple.

For `<canvas>`:

```
var canvas = document.querySelector('canvas');
var video = document.querySelector('video');

// Optional frames per second argument.
var stream = canvas.captureStream(25);
// Set the source of the <video> element to be the stream from the <canvas>.
video.srcObject = stream;
```

For `<video>`:

```
var leftVideo = document.getElementById('leftVideo');
var rightVideo = document.getElementById('rightVideo');

leftVideo.onplay = function() {
  // Set the source of one <video> element to be a stream from another.
  var stream = leftVideo.captureStream();
  rightVideo.srcObject = stream;
};
```

Note: `captureStream()` can only be called after the video element is able to play video; that's the reason it's in the `onplay` handler here.

But why?

The `captureStream()` method makes it possible to record or live stream from canvas and media elements:

- Record and stream game play from a `<canvas>`
- Capture video from a camera, then add additional content or effects
- Create picture-in-picture effects from multiple videos via a `<canvas>`
- Combine video and images (from files or a camera or both) in a `<canvas>`
- Live-stream video played from a file
- Use a recorded audio or video message for a video or voice mail

Essentially, `captureStream()` enables JavaScript to construct and "inject stuff" into a MediaStream.

The small print

- Attempting to use `captureStream()` with a media element that implements content protection via Encrypted Media Extensions will throw an exception.
- When capturing from a `<canvas>`, the maximum frame rate is set when `captureStream()` is called. For example, `canvas.captureStream(10)` means that the canvas outputs between 0 and 10 fps. Nothing gets captured when nothing gets painted on the `<canvas>`, and 10 fps is captured even if the `<canvas>` gets painted at 30 fps. There is a bug with more discussion filed on the `captureStream` spec.

- The dimensions of a `captureStream()` video match the `<canvas>` it was called on.

Demos

Canvas

- Stream from a canvas element to a video element
- Stream from a canvas element to a peer connection

Video

- Stream from a video element to a video element
- Stream from a video element to a peer connection

Support

- Canvas `captureStream()`: Firefox 43 or above; Chrome 50 and above with chrome://flags/#enable-experimental-web-platform-features enabled, or Chrome 52 and above by default.
- Video and audio `captureStream()`: Firefox 47; Chrome 52 and above with chrome://flags/#enable-experimental-web-platform-features enabled, or Chrome 53 and above by default.

Find out more

- Firefox implementation bug
- Chrome Platform Status

Last updated July 2, 2018.