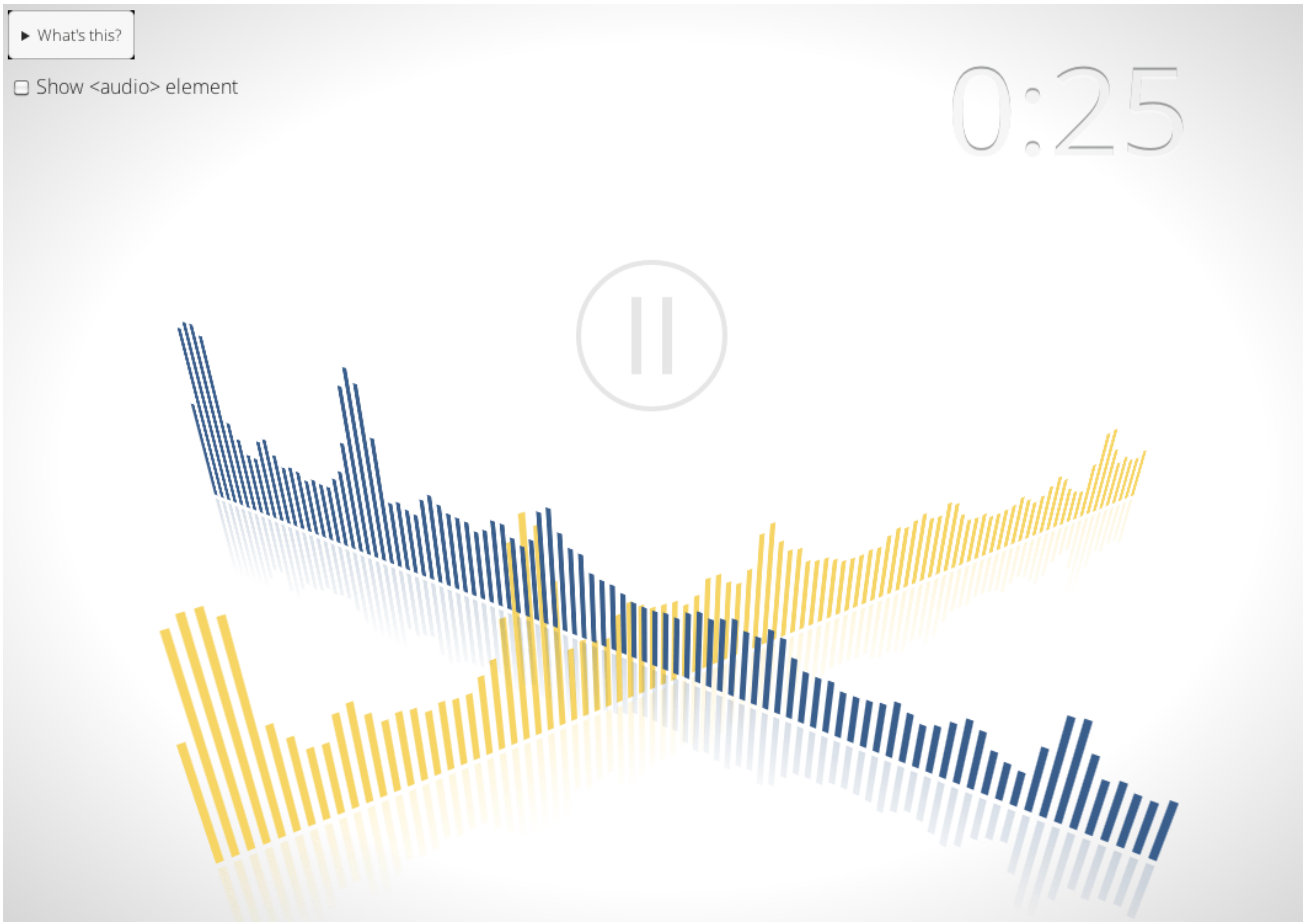
HTML5 audio and the Web Audio API are BFFs!

By Eric Bidelman

Engineer @ Google working on web tooling: Headless Chrome, Puppeteer, Lighthouse



DEMO

As part of the MediaStream Integration with WebRTC, the Web Audio API recently landed an undercover gem known as `createMediaElementSource()`. Basically, it allows you to hook up an HTML5 `<audio>` element as the input source to the API. In layman's terms...you can visualize HTML5 audio, do realtime sound mutations, filtering, etc!

Normally, the Web Audio API works by loading a song via XHR2, file input, whatever,....and you're off. Instead, this hook allows you to combine HTML5 `<audio>` with the visualization, filter, and processing power of the Web Audio API.

Integrating with `<audio>` is ideal for streaming fairly long audio assets. Say your file is 2 hours long. You don't want to decode that entire thing! It's also interesting if you want to

build a high-level media player API (and UI) for play/pause/seek, but wish to apply some additional processing/analysis.

Here's what it looks like:

```
// Create an <audio> element dynamically.
var audio = new Audio();
audio.src = 'myfile.mp3';
audio.controls = true;
audio.autoplay = true;
document.body.appendChild(audio);

var context = new webkitAudioContext();
var analyser = context.createAnalyser();

// Wait for window.onload to fire. See crbug.com/112368
window.addEventListener('load', function(e) {
  // Our <audio> element will be the audio source.
  var source = context.createMediaElementSource(audio);
  source.connect(analyser);
  analyser.connect(context.destination);

  // ...call requestAnimationFrame() and render the analyser's output to canvas.
}, false);
```

As noted in the code, there's a bug that requires the source setup to happen after **`window.onload`**.

The next logical step is to fix crbub.com/112367. Once that puppy is ready, you'll be able to wire up WebRTC (the **`navigator.getUserMedia()`** API in particular) to pipe audio input (e.g mic, mixer, guitar) to an **`<audio>`** tag, then visualize it using the Web Audio API. Mega boom!

Last updated July 2, 2018.