**How to Bundle Audio File into JS App**

By using the [HTML audio element](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/audio), we can embed sound in documents. When your app in running in a server, we can add audio files and refer to the paths in the audio element’s src attribute.

What if you are creating an JS bundle that can be injected into a web site as a script tag? In this post, we will look into how this can be done.

The quickest way is to convert mp3 file into a base64 encoded string, then add the string in HTML. In this way, the bundle can include audio.

I learned this trick from a friend of mine, who also has a blog site called [mydevhack](https://mydevhack.com/). It’s got a lot of cool front end hack. You should check it out.

**Example**

I made a bell that rings. This is a single JavaScript bundle that I am injecting into this div. You can see the base64 encoded string in the bundle [here](https://mydatahack.github.io/ringbell/ring-bell-widget-nonprod.min.js). The source code can be found [here](https://github.com/mydatahack/ring-bell-with-html-audio-example).

<script src="<https://mydatahack.github.io/ringbell/ring-bell-widget-nonprod.min.js>"></script>

**Steps**

1. **Convert mp3 into base64 encoded string**

Once you have an audio file, I recommend you to edit the file with an audio editing tool to trim the unnecessary leading and trailing part and export it with lower quality mp3. You don’t need to have a fancy tool like ProTools or Ableton. You can just download an open source audio editor like [Audacity](https://www.audacityteam.org/).

Compressing audio files is an important step because we want to minimise the size of the bundle. Another technique is to make the audio file as mono instead of stereo to reduce the size. The bigger the file is the bigger the base64 encoded string will be. If you want to go for the quality, It is probably better to have a good quality audio file stored somewhere instead of bundling it.

If you are in a hurry, the compression step doesn’t need to happen. You can just convert mp3 to base64 by using a website offers free mp3 to base64 conversion service like [Base64Guru](https://base64.guru/converter/encode/audio/mp3).

1. **Add the base64 encoded string into HTML**

Once you convert the audio into base64 encoded string, all you need to do is to include it in the source tag inside the audio tag. You can have a constant variable and import it into the part where you want to use the audio

1. **Add JavaScript to play sound**

Then you can simply add JavaScript to trigger play.

Here is an example code with Audio embedded with the function to trigger audio play. Make sure to prepend data:audio/mpeg;base64, to the base64 encoded audio text.

<script src="https://gist.github.com/mydatahack/0ab032dcff9b7b0d1c2a4552dc4d403b.js"></script>