

[Overview](#)[Documentation](#)[Options](#)[Methods](#)[Events](#)[API](#)[Examples](#)[FAQ](#)[Upgrade](#)[Plugins](#)[Projects](#)

Documentation

Getting started

First of all, you need to insert the **wavesurfer.js** library into your HTML page. You can grab the latest version from unpkg.com.

```
<script src="https://unpkg.com/wavesurfer.js"></script>
```

Create a container where you want the waveform to appear:

```
<div id="waveform"></div>
```

Next, in your JavaScript code, create an instance of the global **WaveSurfer** object.

```
var wavesurfer = WaveSurfer.create({
  container: '#waveform'
});
```

If you are using Webpack or another bundler, you can install **wavesurfer.js** with NPM:

```
npm install wavesurfer.js
```

To use the library, you will need to include it from your code using CommonJS:

```
var WaveSurfer = require("wavesurfer.js");
```

Or ES6 syntax:

```
import WaveSurfer from "wavesurfer.js";
```

Parameters

The only required parameter is **container**. It can be either a unique CSS3 selector, or a DOM element.

However, you can also pass any number of **options**. For example, to make the waveform scrollable, pass the the **scrollParent** option:

```
var wavesurfer = WaveSurfer.create({
  container: '#waveform',
  scrollParent: true
});
```

Loading the audio

After creating an instance, you may want to load an audio track and draw its waveform. You can load files either from the same domain:

```
wavesurfer.load('../audio/song.mp3');
```

Or from another server, if it supports **CORS headers**. For example:

```
wavesurfer.load('http://ia902606.us.archive.org/35/items/shortpoetry_047_librivox/song_cjrg_teadale_64kb.mp3');
```

wavesurfer.js will load the file, decode it and display a nice waveform image. When it's done, it will fire the **ready** event.

Listening to events

wavesurfer.js has a number of useful **events** you can subscribe to. The **ready** event, mentioned above, can be used like this:

```
wavesurfer.on('ready', function () {
  wavesurfer.play();
});
```

Calling methods

You can also trigger various actions on the player, such as **wavesurfer.pause()**, **wavesurfer.skipForward()**, **wavesurfer.toggleMute()** etc.

Take a look at the list of all available **methods**.

Creating A Custom Renderer

The provided **Canvas** and **MultiCanvas** renderers should support the vast majority of use cases, but **wavesurfer.js** allows custom renderers to be loaded. You can create your own renderer object (for example, by copying the **Canvas** or **MultiCanvas** renderer code into your own file and changing as necessary). Note that custom renderer objects don't have to be added to the **wavesurfer.js** repository, but you're welcome to raise a pull request if you think others will find your renderer useful.

If you name your own renderer **WaveSurfer.Drawer.MyDrawer**, then you can use the renderer by specifying the **renderer** parameter as **'MyDrawer'**.