

Loading Spinner Customizations

If you've wondered about how to customize that little Kaltura logo loading spinner on the player, then you've come to the right spot. We'll cover some examples of things you can do, but before we dive in, there is some baseline information that will be helpful in understanding what is possible. In this guide we'll cover some basics of using CSS to customize the look and feel of the player. If you need more control than what CSS offers, you can write your own custom plugin, or override the player config with a custom uiComponent in your constructor. For help on the latter, see <https://github.com/kaltura/playkit-js-ui/blob/master/docs/setting-customized-spinner.md>.

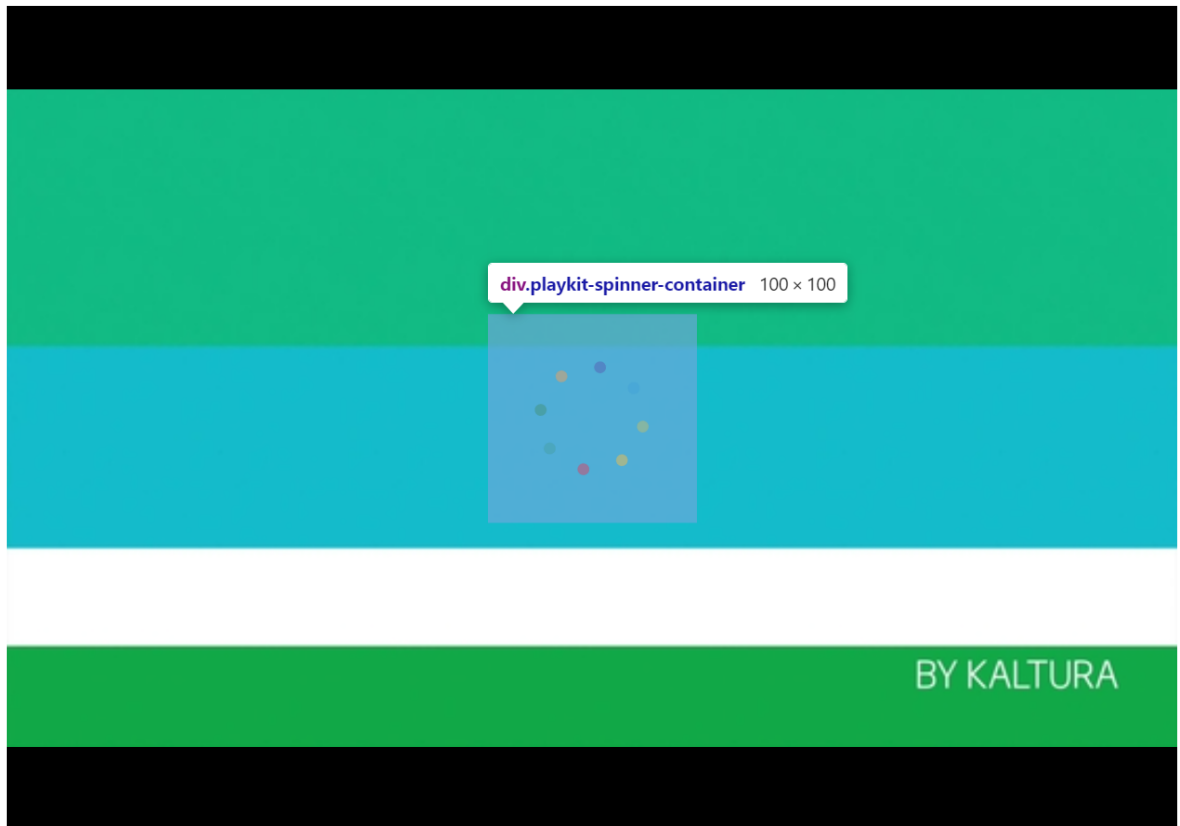
Reference

Before we go about customizing the loading spinner, it's important to understand the underlying HTML elements, and existing CSS classes. The elements consist of a parent `<div>`, child `<div>`, and 8 child ``s.

```

    <div>
      <div class="playkit-player-gui" id="player-gui">
        <div class="playkit-overlay-action" style="position: absolute; inset: 0px;"></div>
        <div class="playkit-video-area" style="position: absolute; inset: 0px;">⋮</div>
        <div class="playkit-gui-area" style="position: absolute; inset: 0px;">flex
          <div style="pointer-events: auto;">
            <div class="playkit-loading-backdrop playkit-show">
              <div class="playkit-spinner-container" style="
                "> == $0
              >
                <div class="playkit-spinner">
                  <span></span>
                  <span></span>
                  <span></span>
                  <span></span>
                  <span></span>
                  <span></span>
                  <span></span>
                  <span></span>
                </div>
              </div>
            </div>
            <div class="overlay-portal" aria-live="polite"></div>
            <div class="playkit-playback-controls playkit-center-playback-controls">⋮</div>
            </div>
            <div class="playkit-top-bar">⋮</div>
            <div class="playkit-interactive-area">
              <div style="pointer-events: auto;">
                <div data-contrib-item="floating-manager" class></div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  
```

- At the top level, we have a `<div>` element with the class "playkit-spinner-container". This is the outer wrapper for the spinner. Note that this wrapper `<div>` doesn't actually spin.



- Under that, we have a child `<div>` element with the class "playkit-spinner". This is the `<div>` that spins.







- And under that child `<div>`, we have 8 child `` elements with no class. These are the spinning dots that you see. Note that there are no classes for these, so if you want to target them to change colors or other attributes independently, you'll need to use `:nth-child` CSS selectors.




Now that you know the elements and classes, you'll also need to know where to configure the player to have it load your custom CSS. In the player studio in the KMC, under the 'Visual' tab,


you'll find the 'External CSS' checkbox and url configuration:


[<](#) **Google Brand Player**  55091672







VISUAL


Player color 

 #006EFA


Live color 

 #E12536

Player background 

 #000000

☒ **External CSS**

CSS URL 

<https://longingforhome.github.io/kaltura/css/8>

After enabling the 'External CSS' checkbox, then enter the url where you are hosting your custom CSS, and Save. NOTE: after saving the config, it can sometimes take up to 10 minutes before the CDN cache where your player is stored refreshes.

Notes on animation

While the animation of the spinner is nice out-of-the-box, there may be times that you'd want to make some alterations, like slowing it down (or speeding it up). The core of the animation is controlled by CSS via the "playkit-spinner" class. The default values are:

```
.playkit-spinner {  
  width: 100px;  
  height: 100px;  
  position: relative;  
  animation: playkit-kaltura-spinner 2.5s infinite;
```

```
animation-duration: 2.5s;
animation-timing-function: ease;
animation-delay: 0s;
animation-iteration-count: infinite;
animation-direction: normal;
animation-fill-mode: none;
animation-play-state: running;
animation-name: playkit-kaltura-spinner;
animation-timeline: auto;
animation-range-start: normal;
animation-range-end: normal;
}
```

If you want to slow the spinner down, just override the "animation-duration" value and lengthen the duration (the longer the value, the slower the spin):

```
.playkit-spinner {
  animation-duration: 5s !important;
}
```

Or conversely, if you want to speed it up, shorten the duration:

```
.playkit-spinner {
  animation-duration: 1.25s !important;
}
```

You can change any of the values to effect how the animation renders. For more info on CSS animation values, see <https://developer.mozilla.org/en-US/docs/Web/CSS/animation>.

Examples

Now that you understand the basic structure, function, and how to configure, then we can go about customizing.

Google Spinner Example

Let's create an example where we set the spinner background to have the Google logo, and change the spinner dots to white. To do this, we'll need to set the background of the parent wrapper `<div>` (remember, that's the one that does NOT spin). Then we'll just change the color of the ``s to white. Your CSS will look like this:

```
div.playkit-spinner > span {
  background-color: white !important; /* can change to whatever hex/rgb value
you want for the color of the dots */
}
```

```
div.playkit-spinner-container {
  background-image:
url(https://cfvod.kaltura.com/p/5954112/sp/595411200/raw/entry_id/1_ysx0ezhr/versi
on/100001); /* change this url to whatever hosted image you want to use.  in this
example, the image is 64x64 px */
  background-repeat: no-repeat;
  background-position: center;
}
```

This should render something like this:



McDonalds Spinner Example

Let's take the Google example, and tweak it just a little to make the spinner dots alternating colors. We'll just need to make use of the :nth-child selector, and add specifications for odd and even children (to alternate the colors):

```
div.playkit-spinner > span:nth-child(even) {
  background-color: #EA0000 !important; /* will target the even spans, so
2,4,6,8 */
}

div.playkit-spinner > span:nth-child(odd) {
  background-color: #FFC300 !important; /* will target the odd spans, so 1,3,5,7
*/
}

div.playkit-spinner-container {
  background-image:
url(https://cfvod.kaltura.com/p/5954112/sp/595411200/raw/entry_id/1_n3ixrm3t/versi
on/100001); /* change this url to whatever hosted image you want to use.  in this
```

```
example, the image is 50x50 px */
background-repeat: no-repeat;
background-position: center;
}
```

This should render something like this:



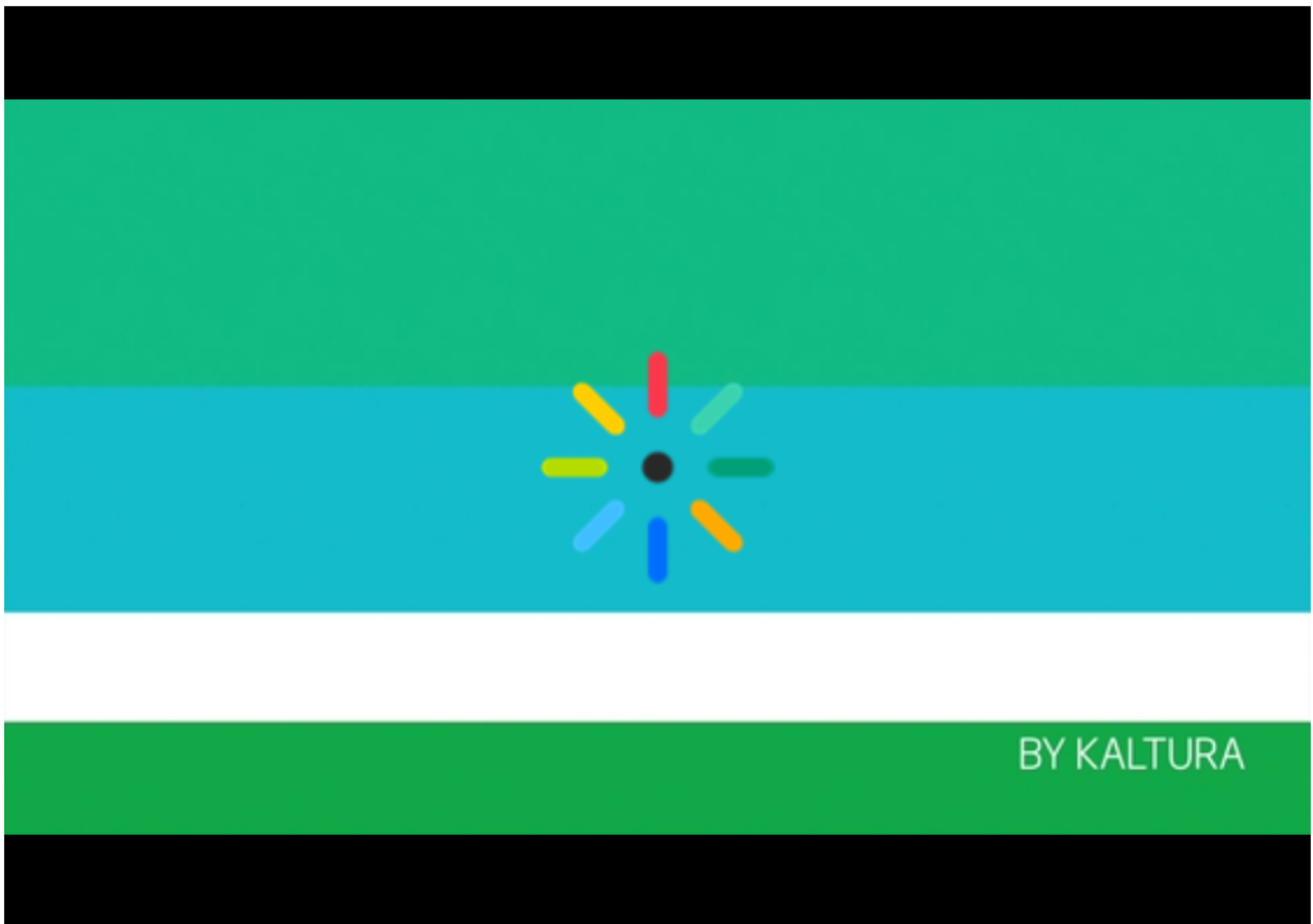
Simple Image Loader Example

Let's say that you don't want the spinner at all and would prefer to simply leverage your own image. For this, we'll want to set your image as the background, and then just hide the spinner spans completely:

```
div.playkit-spinner > span {
  display: none; /* hide all the children spans */
}

div.playkit-spinner-container {
  background-image:
url(https://cfvod.kaltura.com/p/5954112/sp/595411200/raw/entry_id/1_8k73p2ae/version/100001); /* change this url to whatever hosted image you want to use. in this
example, the image is 100x100 px */
  background-repeat: no-repeat;
  background-position: center;
}
```

This should render something like this:



Rotating Image Loader Example

The last example was nice and easy, but maybe you want a little more action. Well, since the image we used can be easily rotated without impacting any readability, how about we just have it spin? To make it easy, since the `<div>` with the "playkit-spinner" class already rotates out-of-the-box, then let's just take advantage of that:

```
div.playkit-spinner > span {
  display: none; /* hide all the children spans */
}

div.playkit-spinner {
  background-image:
url(https://cfvod.kaltura.com/p/5954112/sp/595411200/raw/entry_id/1_8k73p2ae/version/100001); /* change this url to whatever hosted image you want to use. in this
example, the image is 100x100 px */
  background-repeat: no-repeat;
  background-position: center;
}
```


This should render something like this:



Wrapping Up

Hopefully this guide and examples have given you some direction and inspiration on customizing the loading spinner of your Kaltura player.

Caveats/Notes

Things worth mentioning as you embark on this journey:

- When using externally hosted CSS for customizations, the player tries to asynchronously load the CSS when it's rendering. If the spinner renders before the CSS finishes loading, then you may see a short blip of the original spinner until the CSS is loaded and applied.