



Categories

CSS

DevWall

IDEs

Resources

HTML

JavaScript

Laravel

NodeJS

PHP

GraphQL

Sponsored

Privacy Policy

Contact Us

About

Courses

Beginner Data Visualisations Course



JANUARY 27, 2020

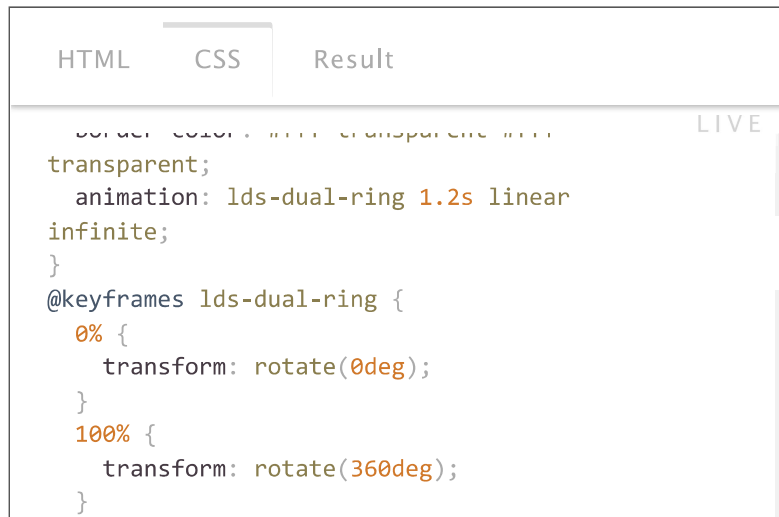
jQuery AJAX Loading Spinners Tutorial with Examples

Loading spinners are awesome, and there are a lot of them available from a variety of sources. One of these is loading.io, a service that helps you create loading animations with icons, GIF's, SVG's and even backgrounds.

If you've ever or are currently working with a highly-dynamic data analysis web project, then you will of come to know that requesting data via AJAX is very common. Now, rather than just have some simple 'Loading' message presented to the user when a button is clicked, for instance, we can show something much nicer, a spinner.

In this tutorial, we will add one of the loading.io icon spinners to an ajax call, so that the user can visually see that the process is happening during a request.

First of all, let's pick a spinner, I chose the one below –



These spinners are EXTREMELY easy to set up, all you need to do is copy the provided CSS into your stylesheet and copy the HTML tag provided into your HTML, that is it!

For the example I've used, here is the CSS required –

```
.lds-dual-ring {
  display: inline-block;
  width: 80px;
  height: 80px;
}
.lds-dual-ring:after {
  content: " ";
  display: block;
  width: 64px;
  height: 64px;
  margin: 8px;
  border-radius: 50%;
  border: 6px solid #fff;
}
```

```
... border-color: #fff transparent #fff transparent;
... animation: lds-dual-ring 1.2s linear infinite;
}
@keyframes lds-dual-ring {
... 0% {
...   transform: rotate(0deg);
... }
... 100% {
...   transform: rotate(360deg);
... }
}
```

And the HTML

```
<div class="lds-dual-ring"></div>
```

And boom, you've got yourself a spinner, All in a minutes work. I dare even say next to zero coding experience is needed!

Next up, we need some data, and I've found a pretty simple API that allows you to call it without any API token. It's data from the Forbes rich list so we can request the top 400 richest people in the world. – Check it out using the URL – <https://forbes400.herokuapp.com/api/forbes400?limit=400>

So, now we have a spinner, some data, alls is left to do is add a button to click and some JQuery to handle the button click and subsequently execute the AJAX call.

Step 1

Add the following HTML to your .html file.

```
<button id='getDataBtn'>Get Data</button>

<div id="richList"></div>

<div id="loader" class="lds-dual-ring hidden overlay"></div>
```

Notes:

- Get Data button will be listened for via JQuery and execute the Ajax once pressed.
- The richList div element will be used to put in the list of people returned from the API.
- The loader element will sit hidden away (note: hidden class added) prepared to be brought into view. Also, note the overlay class which I will explain later.

Step 2

Add the following CSS between some <style> tags or into your CSS file –

```
body {
    background: #ececec;
}

/*Hidden class for adding and removing*/
.lds-dual-ring.hidden {
    display: none;
}

/*Add an overlay to the entire page blocking any further presses to bu
.overlay {
    position: fixed;
    top: 0;
    left: 0;
    width: 100%;
```

```

..... height: 100vh;
..... background: rgba(0,0,0,.8);
..... z-index: 999;
..... opacity: 1;
..... transition: all 0.5s;
..... }
.....
..... /*Spinner Styles*/
..... .lds-dual-ring {
.....     display: inline-block;
.....     width: 80px;
.....     height: 80px;
..... }
..... .lds-dual-ring:after {
.....     content: " ";
.....     display: block;
.....     width: 64px;
.....     height: 64px;
.....     margin: 5% auto;
.....     border-radius: 50%;
.....     border: 6px solid #fff;
.....     border-color: #fff transparent #fff transparent;
.....     animation: lds-dual-ring 1.2s linear infinite;
..... }
..... @keyframes lds-dual-ring {
.....     0% {
.....         transform: rotate(0deg);
.....     }
.....     100% {
.....         transform: rotate(360deg);
.....     }
..... }
..... }

```

Notes:

- The `.hidden` class will be used for the spinner as a default loading state so that the user doesn't see it. We toggle this on and off with JQuery.
- The `.overlay` class will add a full-screen overlay with the spinner and on top of any other on-screen elements. This

prevents users from clicking anything else during the load stage. Always a good idea!

- The rest of the styles are required for the spinner itself.

Step 3

Add the following JavaScript to your .html page or in its own script file.

```
// Define our button click listener
.... $('#getDataBtn').click(function () {

    .... // On click, execute the ajax call.
    .... $.ajax({
    ....     type: "GET",
    ....     url: "https://forbes400.herokuapp.com/api/forbes400?limit=400"
    ....     dataType: 'json',
    ....     beforeSend: function () { // Before we send the request, remov
    ....         $('#loader').removeClass('hidden')
    ....     },
    ....     success: function (data) {
    ....         // On Success, build our rich list up and append it to the
    ....         if (data.length > 0) {
    ....             let richList = "<ol>";
    ....             for (let i = 0; i < data.length; i++) {
    ....                 console.log(data[i].uri);
    ....                 richList += "<li>" + data[i].uri + "</li>";
    ....             }
    ....             richList += "</ol>"
    ....             $('#richList').html(richList);
    ....         }
    ....     },
    ....     complete: function () { // Set our complete callback, adding t
    ....         $('#loader').addClass('hidden')
    ....     },
    .... });
    .... });
```

Notes:

- We define a click handler to listen for button clicks

- Once the button is clicked we start the Ajax functionality
- We define the URL and data type and then use the beforeSend function to show our spinner by removing the .hidden class from the spinner element. Now the spinner is overlaid and spinning the user can see the request is in progress and can't interrupt it by clicking anything else.
- With the success function, we build up our HTML from the API response, building a numbered list with the names of all the people on the API list. Then we append the HTML to the #richList element.
- In the complete callback function, we finally add the .hidden class back to the spinner, hiding it away until next required.

Now, if we put this all together within a CodePen the result will be as the following –

Hit the GetData button to request the data and see the spinner in action.



Summary

Using this ten-minute tutorial you can totally transform your user's experience when requesting data, whether it be to add new data to a data table, download an excel file or load data into a chart. The method of

overlaying with CSS also brings a new element of preventing any unwanted further clicks on the button or other elements. Of course, you can handle this in different ways but this is a quick and simple method.

Loading.io has plenty of spinners and other loading animations and is well worth looking at. [Check them out here](#).



The Author: Dan Englishby

[View all posts by Dan Englishby.](#)

[@DanEnglishby](#)

Posted in [CSS](#) , [DevWall](#) , [JavaScript](#)

Previous:

[Private URL Shortener With PHP Tutorial](#)

Next:

[Building Data Visualizations in PowerPoint Using](#)

[Python](#)

1 thought on “jQuery AJAX Loading Spinners Tutorial with Examples”

Aang Kurts says:
March 15, 2021 at 2:13 pm

Nice and simple. Thank you!

Reply

Leave a Reply

Comment *

Name

Post Comment

Search ...

Copyright 2022 Code Wall. All rights reserved.