Hi fellow developers, a couple of weeks I published an article featuring the weather app challenge from Free Code Camp were I had a lot of fun, if you would like to check it out click here (https://www.linkedin.com/pulse/check-out-my-freecodecamp-weather-app-waldo-lavaut/) [link].

Today’s lesson is quite simple, we’re building a random quotes app, another challenge from FreeCodeCamp, the requirements are:

1. I can click a button to show a new random quote
2. I can click another button to tweet out that quote

For this we will only use an index.html file and a JS file, no CSS file since the customization is minimal and Bootstrap comes loaded with a lot of cool CSS stuff!

Let’s grab the starter html template from Bootstrap 4:

* Example -

We’re going to use in the <body></body> a nice card that will contain a header with a tweet icon to be able to tweet that quote, as well as a card body with the quote text and a footer. One very important thing is that you need to use the minified version of jQuery instead of the slim version that the bootstrap 4 template have, as of jQuery 3.3.1, the slim build does not include the ajax and effects modules, not allowing us to make any sort of http methods using jQuery (you can still do it using vanilla JavaScript though), curious about what I just said, check out the jQuery docs and read the slim build paragraph here (https://blog.jquery.com/2016/09/22/jquery-3-1-1-released/)[link]

Notice that I’m also using the font awesome icons, so I’m including a link to their CDN file on the head of the document. If you don’t know font awesome, they are absolutely amazing, learn more about them here (https://fontawesome.com/) [link].

The card header has also some inline CSS to use “display: flex;” and be able to display the header title to the left and the tweeter icon to the right using the “justify-content: space-between”, that comes from flexbox CSS, here a nice guide (https://css-tricks.com/snippets/css/a-guide-to-flexbox/) [link]. By the way, you should also learn CSS Grid Layout since it operates in both axis (https://css-tricks.com/snippets/css/complete-guide-grid/) [link]. Another thing to notice is that I pre-populated the card body with a “default quote” before hitting the API. All those details on the HTML are trivial for advanced developers but I think it’s worth to mention since it really helps people learning to code.

Ok, now let’s see what we are doing with the JavaScript file. Create a JS file and link it to your HTML (see end of snippet above). The guys at <http://forismatic.com/en/> kindly provided an API that returns random quotes depending on the query sent to their servers, so let’s look at the URI resource and analyze it part by part:

https://api.forismatic.com/api/1.0/?method=getQuote&lang=en&format=jsonp&jsonp=?

According to their short API docs (<http://forismatic.com/en/api/>)

The method part or the URL is the method name to invoke, the Lang should be set to “EN” for English since the default is Russian, and the format we are requesting is JSONP which stands for JSON with Padding, the good thing about JSONP is that enables the sharing of data between different domains without worrying about cross-domain issues which gave me a really hard time with other APIs, this article explain more in detail what the CORS( Cross-Domain-Resource-Sharing) (https://www.html5rocks.com/en/tutorials/cors/)[here].

We start our JS file with the **document.ready** function that will execute the JS file after the HTML is fully loaded, then we create an event listener to allow users tweet the “default” quote that I pre-wrote on the card-body back on the HTML file.

* Example -

Next, I am creating the **getRandomQuote** function that will make a get request to the forismatic API and it will return a random quote in json format per as our URI request above, we will inject that data into our nice bootstrap card:

-Example-

Next, I am checking if the quote Author is missing, since I noticed in some of those.

* Example -

And finally, I create a **tweetQuote** function that will take the quote and it will append it to the twitter share URL. Notice that when you click the tweeter icon we’ll use window.open() to go to the tweeter website and make a new tweet. You should check out the Tweeter API’s docs (https://dev.twitter.com/web/tweet-button) [here].

-Example-

Interestingly the jQuery docs do not specify any error handling for JSONP requests, but here’s a library feature packed for those adventurous devs (https://github.com/jaubourg/jquery-jsonp) [here]. Also, some stack-**overflow** discussion about it: (<https://stackoverflow.com/questions/19035557/jsonp-request-error-handling>) [here].

Here’s the full JS file all wrapper together:

* Example -

That was my approach or implementation for this challenge. Another solution I’ve seen fellow free code campers using is:

1. Grab the formatted JSON data from the API and store it on an array, assuming you get 15 or 20 or 50 different random quotes.
2. Use the built in Math.random() \* the length of the quotes array wrapped in Math.floor() and use that number to get a random quote every time you press the next button, like this:

* Example –

Here’s the HTML for this second example in case you’re wondering:

* Example –

Well folks that’s all today, hope you had fun with this challenge and looking forward to see you reading my next article, until then: Happy Coding!

Waldo Lavaut

Full Stack Web Developer