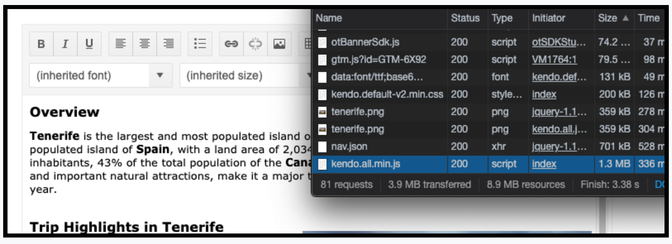
STRIVEN EDITOR

10/27/2019 - Henry Unite

INTRODUCTION

At Striven, we were looking for potential editors that would be suitable for our client side customer portal revamp. In the past, we have always used the Kendo UI editor. We were dynamically loading these components into our pages, but even then the editor was shipping over 1MB of scripts over the network.

One of the key features of this portal was its lightweight optimization. When you look at the Kendo UI minified script over the network, you'll notice a whoping 1.3MB are being shipped to the browser. With the editor component alone making up nearly 1MB of that script.



An editor was a fundamental component of this customer portal, so we wanted to provide a more optimal solution to our users. I was tasked with the research of finding a lighter editor with just as much, or as much as we needed, functionality.

These were some notable candidates:

- Quill
- Froala
- TineMCE

QUILL

Quill would be great; its open source, inline, and used by top companies. My experience when trying to integrate with the editor invloved the developer push for the adoption of the editor's delta api.

My advice to anyone trying to include this control in their project is that you'll have a much easier time designing your system with the concept of delta in mind than trying to bring the concept of the quill editor and its delta api into an existing system.

FROALA AND TINYMCE

So these editors are obviously top tier editors, but usually have some licensing and enterprise support costs that we weren't ready to explore just yet. I never tried to demo or pitch it to my team, but I still think these were candidates worth considering when all else failed.

UNDERTAKING A CUSTOM BUILT EDITOR

After working on mentions and exploring all these different editor solutions, I formed an understanding of the underlying concepts that went into the inner workings of a WYSIWYG editor. My manager was convinced that I was capable of taking on the project of building a tailored editor for Striven, as it would be worth the time and investment to work on this component in house.

We decided to work on the component in phases.

What did we need out of the initial phase of the control?

- Simple Editing Functions (bold, italic, underline, unordered lists)
- File Attachments and Link Insertions
- Mention Support

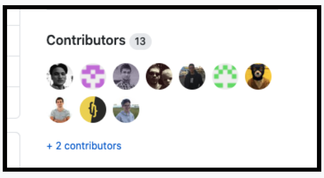
With these functionalities in pipeline, editor phase one development was underway. But I decided that I wasn't going to do it alone.

THE BENEFITS OF OPEN SOURCE

With permission from my manager, I decided that open sourcing this control would have the following benefits:

- Faster development
- Development guidance from the community on building an editor
- Product brand awareness
- Component development independent from the internal system

The idea was a success*, as I could gain input from the open source community and work on the component without having to rely on the internal structure of our client side ecosystem. I also learned a lot as a project maintainer and was really proud of the overall traction that the editor had received in early development.



*Even though 13 contributors isn't a terribly significant amount, I still take pride in it 😅

WHAT I LEARNED FROM THIS PROJECT

I've been actively maintaining this code base for about a year now and if I could given some wisdom to the young software engineer that does it next, this would be the advice I would give:

Avoid document.execCommand

- It's old, buggy, and just a pain of an API to use
- It's one of those Internet Explorer that still lives to see modern web development

Start with an engine or library

- I've explored ways I would've redesigned this component on the trix engine
- I've considered how nice it would've been to use `contenteditable` libraries

Understand the Range API

- Although I don't see this system going anywhere in future iterations of the browser, I still believe it to be mediocre
- Learning how to use the Range API and `window.getSelection()` will make your life much easier
- Try exploring libraries like rangy

FINAL THOUGHTS

When I reflect on the development of this project, my goal was always to accomplish two things:

- Allow developers to contribute, collaborate, and learn open source with a smaller scale project
- It's an editor for Striven, not for anything else (but feel free to use it)

It's satisfying to have the ability of opening an issue and letting someone contribute to this project. I also enjoy the ability to work on this component in an independent environment from Striven. It lets me flex my ES6 muscles and have all the luxuries of hot reloading, webpack loaders, and working in Vue.

There are definitely things that I would've done differently, but at the end of the day it does what Striven needs.