

# Whither Wordpress?

## JAMStack CMS are ready to shine!

By Brian Rinaldi

@remotesynth



STACKBIT

# FLASHBACK CONFERENCE

Celebrating web  
development of today  
and how we got here



February 10th-11th, 2020 • Orlando, Florida • [flashback.dev](https://flashback.dev)

I don't  
hate  
Wordpress



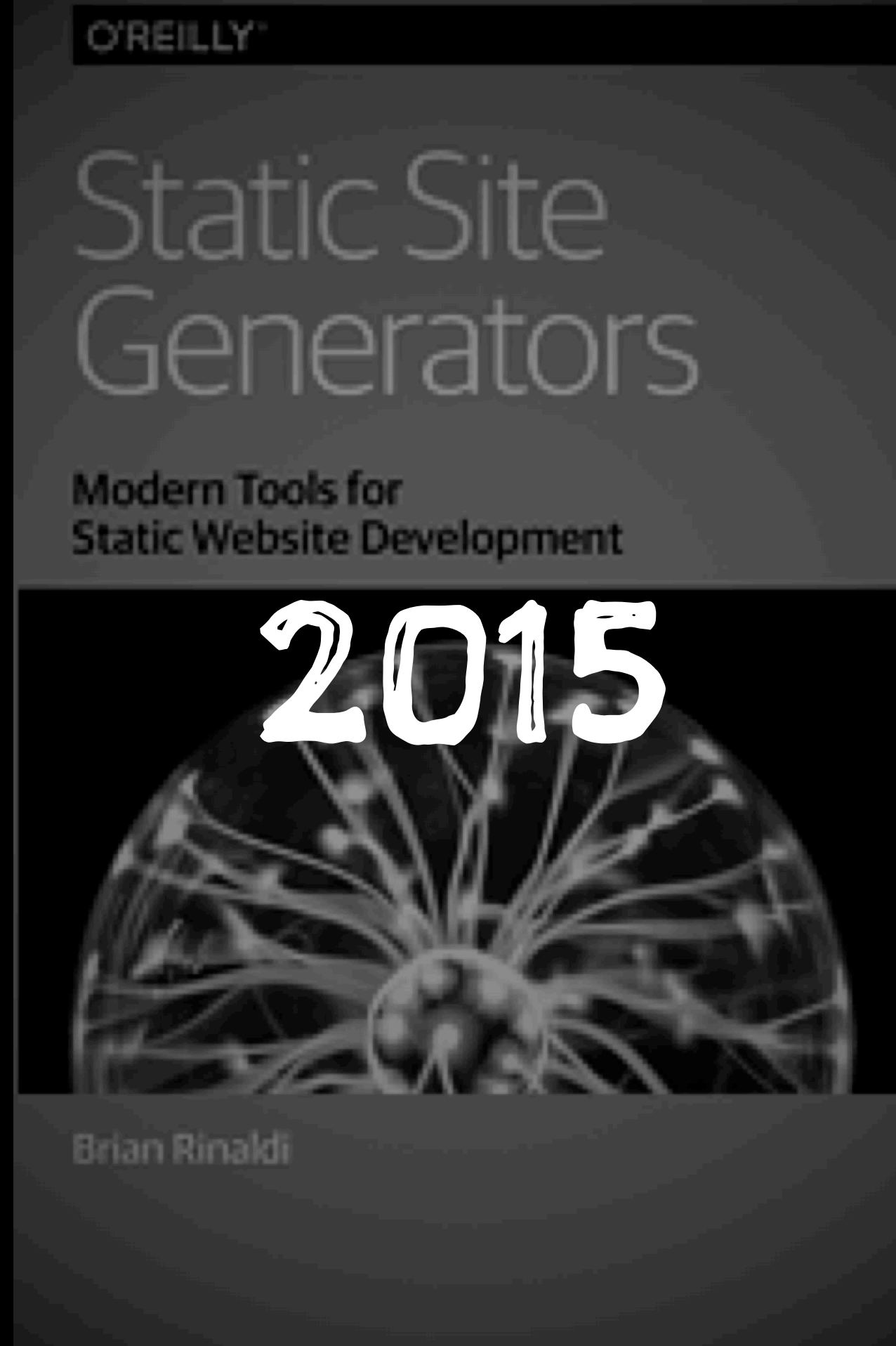
# Comparing Static Site Engines

2015

with Brian Rinaldi



I was a believer in static site tools - or what we now call JAMStack - since the early days. I've been writing and speaking on this topic since at least 2014.



I wrote a report for O'Reilly back in 2015. Back then the process was very labor intensive.

Just in case this isn't already clear, I want to emphasize that static site generators are built for developers. This starts with the development of the site all the way through to adding content. It's unlikely that nondevelopers will feel comfortable writing in Markdown with YAML or JSON front matter, which is the metadata contained at the beginning of most static site engine content or files. Nor would non-technical users likely feel comfortable editing YAML or JSON data files.



Me (Static Site Generators report for O'Reilly 2015)



A LITTLE  
LATER...

Flash forward a couple years  
later things had minimally  
improved



O'REILLY®

# Working with Static Sites

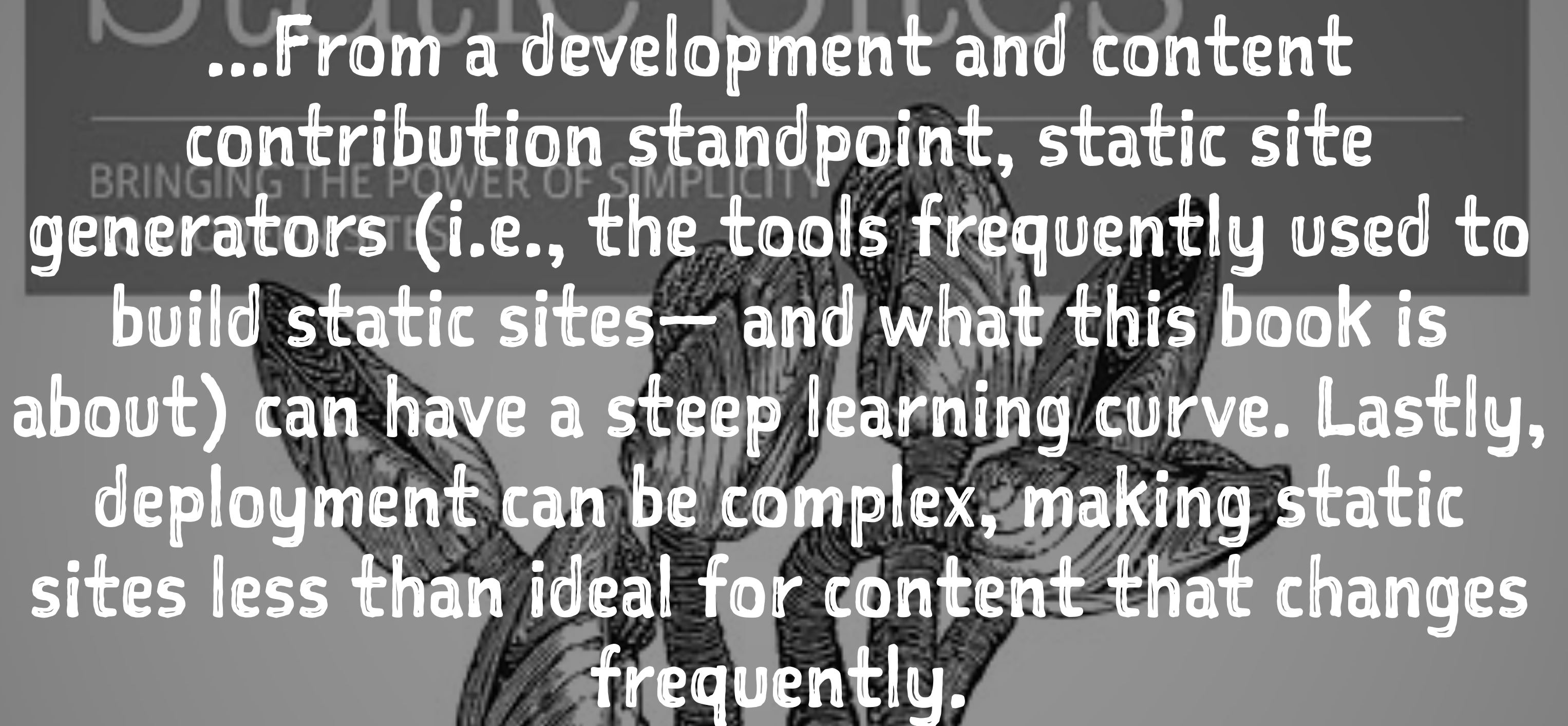
BRINGING THE POWER OF SIMPLICITY  
TO MODERN SITES

2017



Raymond Camden  
& Brian Rinaldi



The background image is a book cover for 'Working with Static Sites' by Me. The title 'Static Sites' is at the top in a large, light-colored serif font. Below it, the subtitle 'BRINGING THE POWER OF SIMPLICITY TO STATIC SITES' is in a smaller, all-caps sans-serif font. The author's name 'Me' is at the bottom. The cover features a detailed, dark-toned illustration of a person's hands and arms, possibly working on a computer or a craft. The text of the quote is overlaid on the upper half of the cover in a white, monospaced font.

...From a development and content contribution standpoint, static site generators (i.e., the tools frequently used to build static sites— and what this book is about) can have a steep learning curve. Lastly, deployment can be complex, making static sites less than ideal for content that changes frequently.

Me (Working with Static Sites, 2017)

The tooling was very primitive. It required a comfort with the command line and knowledge of things like Markdown, YAML and "front matter."

# The Developer Experience Improved

The developer experience started to improve - eliminating the need for manual builds and FTP deployment, for example.



**GitHub** Pages

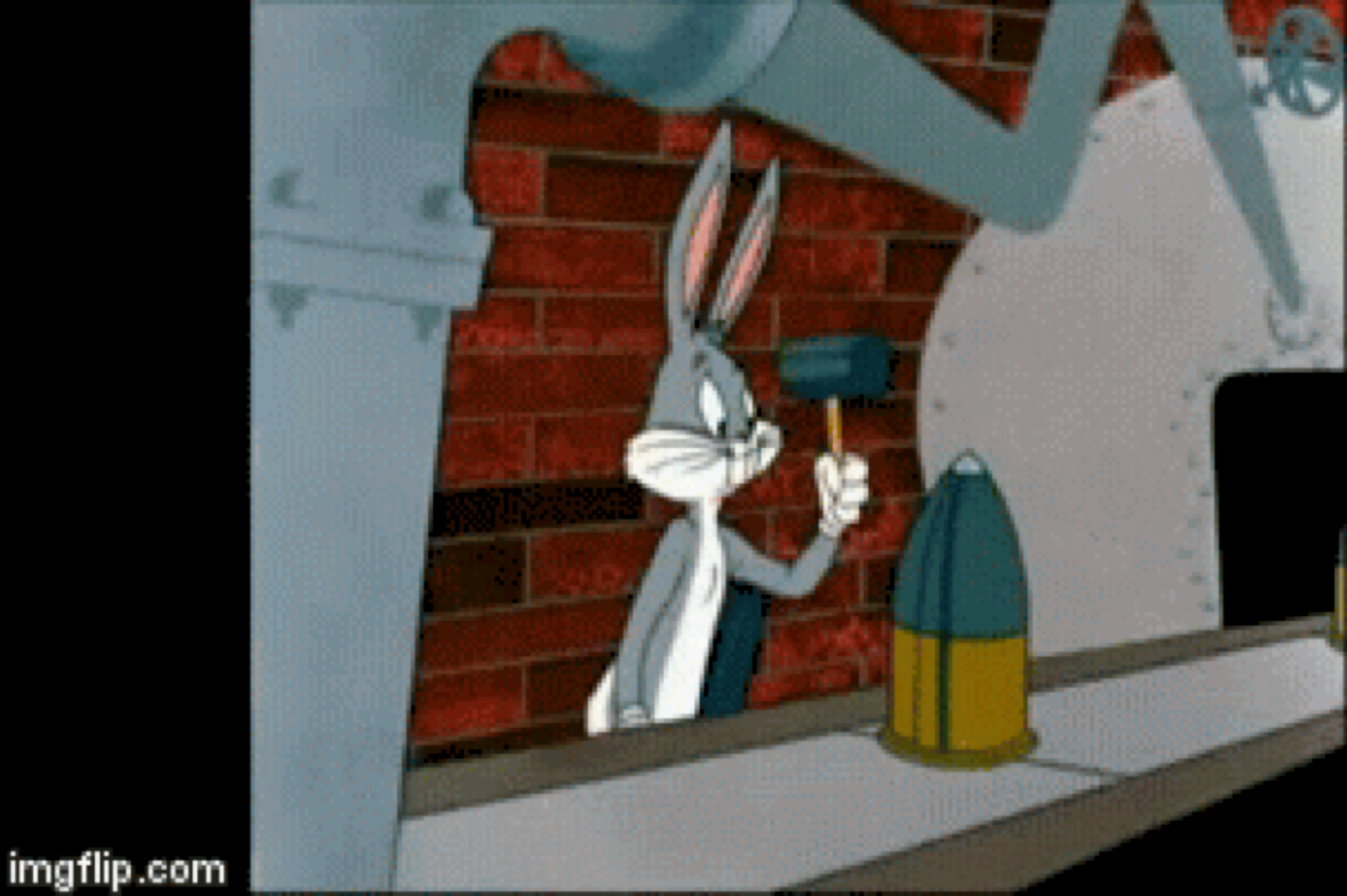
<https://pages.github.com>

GitHub Pages added Jekyll support as far back as 2014 meaning you could just check in your code in GitHub and the page would build automatically.



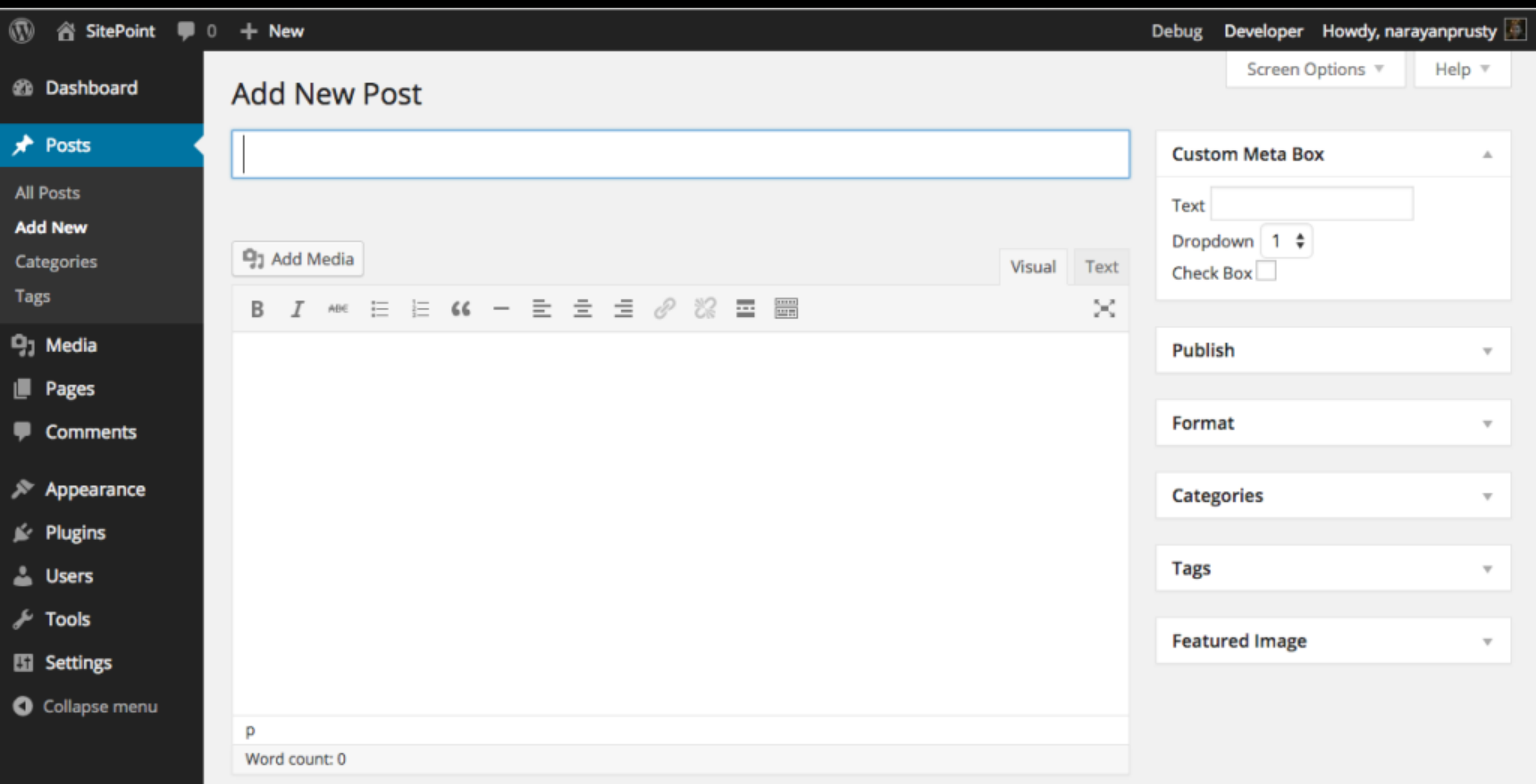
Netlify launched in 2015, these improved the dev experience but didn't impact the content writer/editor experience at all.





The actual process of creating and managing content, however, was still a very manual process of creating and editing text files in the file system.



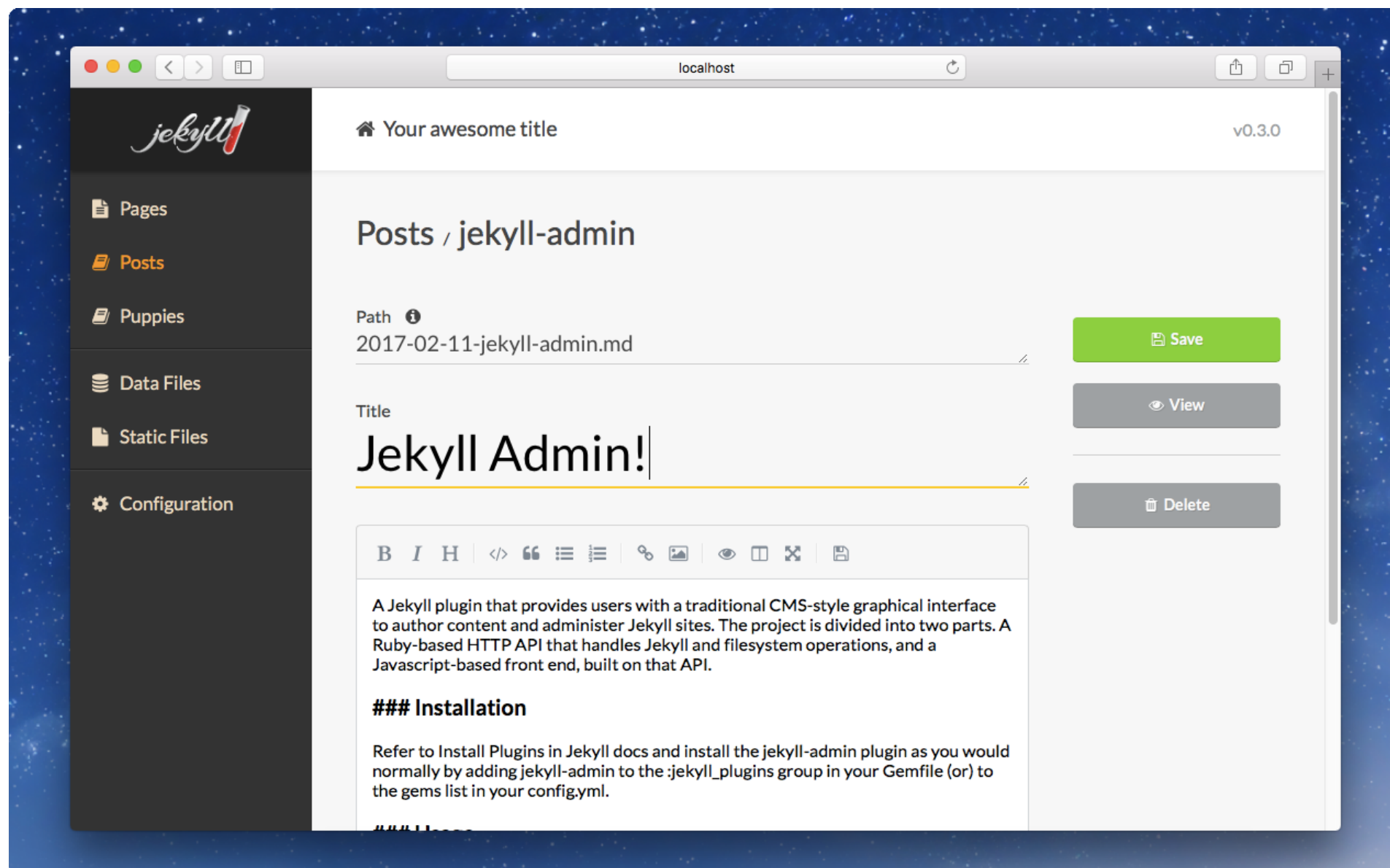


Content creators were used to a WYSIWYG experience and more advanced content management tooling similar to Wordpress site admin

# Static Site CMS

Some solutions tried to resolve this by creating a CMS-like experience but in a static context

# Jekyll Admin



Jekyll is one of the oldest and most widely used static site generators and it was the first that I was aware of to release an officially endorsed admin

# Jekyll Admin

Available at: <https://github.com/jekyll/jekyll-admin>

Installation:

1. Modify your site's Gemfile to add:

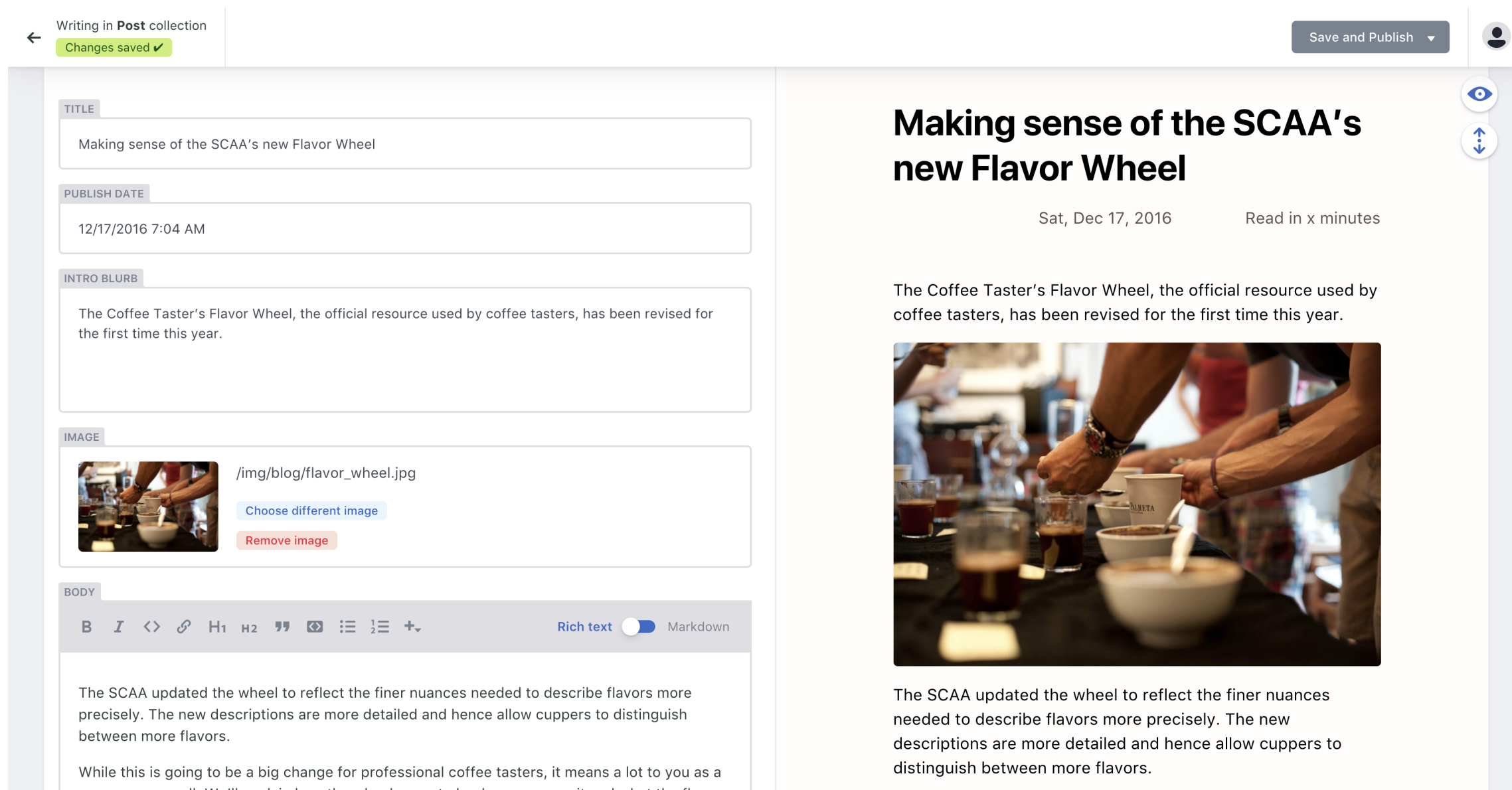
```
gem 'jekyll-admin', group: :jekyll_plugins
```

2. Run `bundle install`

3. Start Jekyll and go to `http://localhost:4000/admin`

Jekyll admin works fine but it is intended to be run locally, so there are no user authentications and there are very limited configuration options

# Netlify CMS



Netlify CMS is an open source project from Netlify that is a highly configurable administration panel that can be added to pretty much any JAMStack site. Unlike Jekyll admin, it is designed to be run publicly – it defaults to Netlify's Identity service but it can be configured to use external OAuth providers for authentication.



# Netlify CMS

Available at: <https://www.netlifycms.org>

Installation:

1. Option A: One Click Install
2. Option B: Custom Install

Installation of NetlifyCMS can be extremely simple assuming this is a new project.

Configuration for existing or highly customized projects can get fairly complex.

# Netlify CMS - Installation

## Start with a Template

[Edit this page](#)

You can add Netlify CMS [to an existing site](#), but the quickest way to get started is with a template. Found below, our featured templates deploy a bare-bones site and Netlify CMS to Netlify ([what's the difference, you ask?](#)), giving you a fully working CMS-enabled site with just a few clicks.



Hugo Site Starter

 Deploy to netlify



Gatsby Site Starter

 Deploy to netlify



Middleman Site Starter

 Deploy to netlify


Just click the button and you are ready to go. However, you are limited to just Hugo, Gatsby or Middleman and deployed to Netlify. Stackbit also supports a very similar installation of NetlifyCMS that works with Hugo, Gatsby and Jekyll.

# Netlify CMS - Installation

1. Add an admin folder with an index.html that includes the Netlify CMS scripts.
2. Add a config.yaml to the admin folder with configuration options and a data specification

# Netlify CMS - Resources

1. Overview and one-click install walkthrough: [A Fresh Look at Netlify CMS Pt. 1](#)
2. Custom install walkthrough: [A Fresh Look at Netlify CMS Pt. 2](#)



# The Headless CMS

Around the same time that JAMStack was taking off the idea of headless CMS began to rise. Contentful, Forestry, Sanity, Dato are a few well known examples.





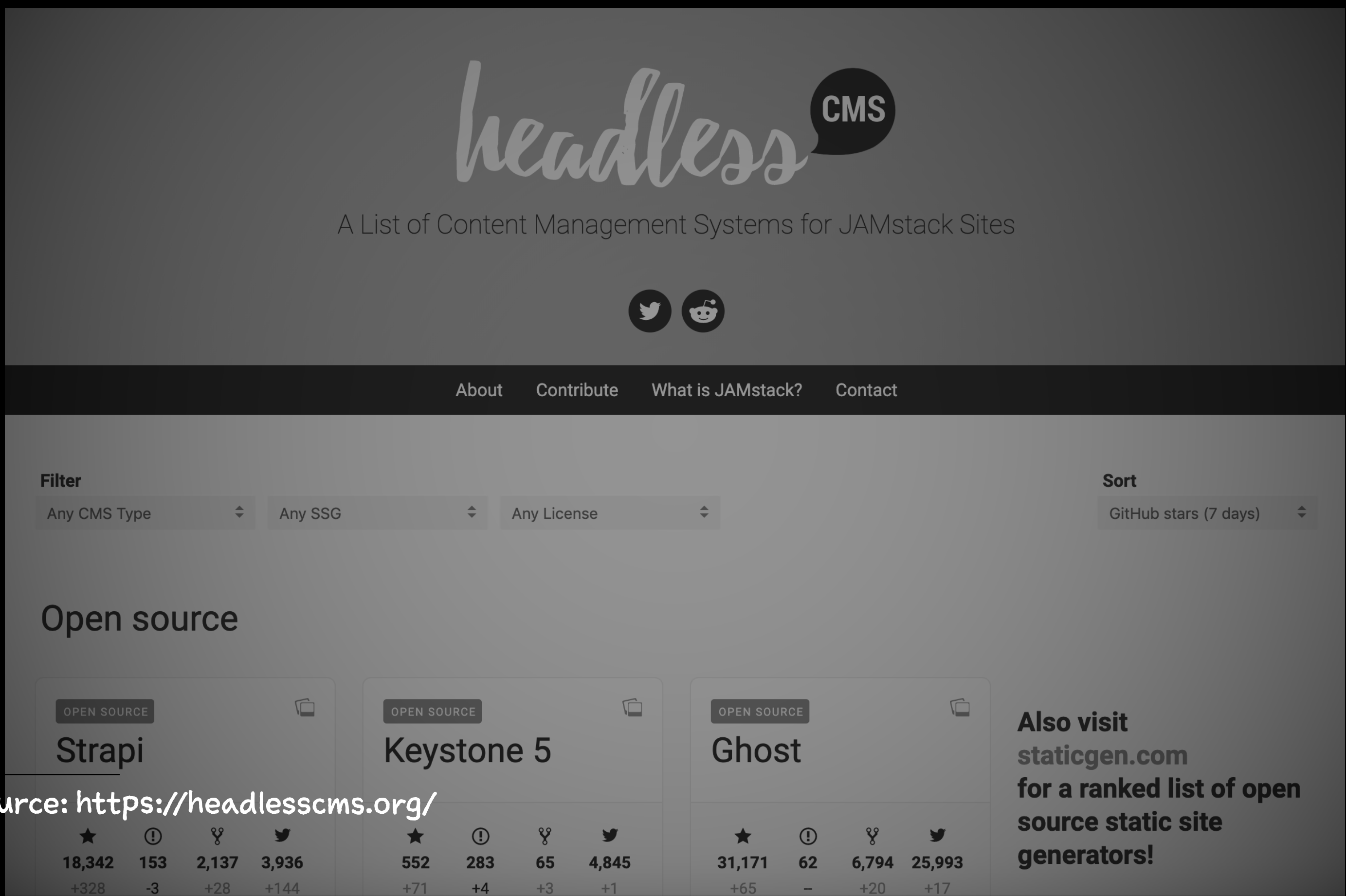
# What is a Headless CMS?

For those of you unfamiliar with the term, a headless CMS is essentially the back end of a CMS decoupled from the presentation of the content on the front-end. A headless-CMS is essentially a content API for the front end but gives the back end content editors the tools they need to create a manage content. Let's look at two examples of different types of headless CMS.

Source: <https://headlesscms.org/>

Service	Headless?	Hosting	Notes
Contentful	Yes	Cloud	A classic headless CMS
Sanity			JSON data structure, accessed via APIs, custom admin area is self-hosted
Cockpit		Self	Comes with admin UI
Strapi			
KeystoneJS			All code, not even an admin UI
WordPress	Sorta – Usually used with head	Self or Cloud	Has a head, but you don't have to use it, you choose to only use APIs to access content if you wish.
Drupal		Self	
CraftCMS		Self	Specifically has a headless mode and GraphQL API. Craft Cloud will bring a cloud-hosted headless variant
NetlifyCMS	Sorta - Doesn't actually store content, just helps edit it.	Cloud	GUI for Git-hosted Markdown
Forestry			
Joomla	No	Self	A classic headed CMS
WordPress		Cloud	Site builder, meant to build hosted/headed sites
Wix			

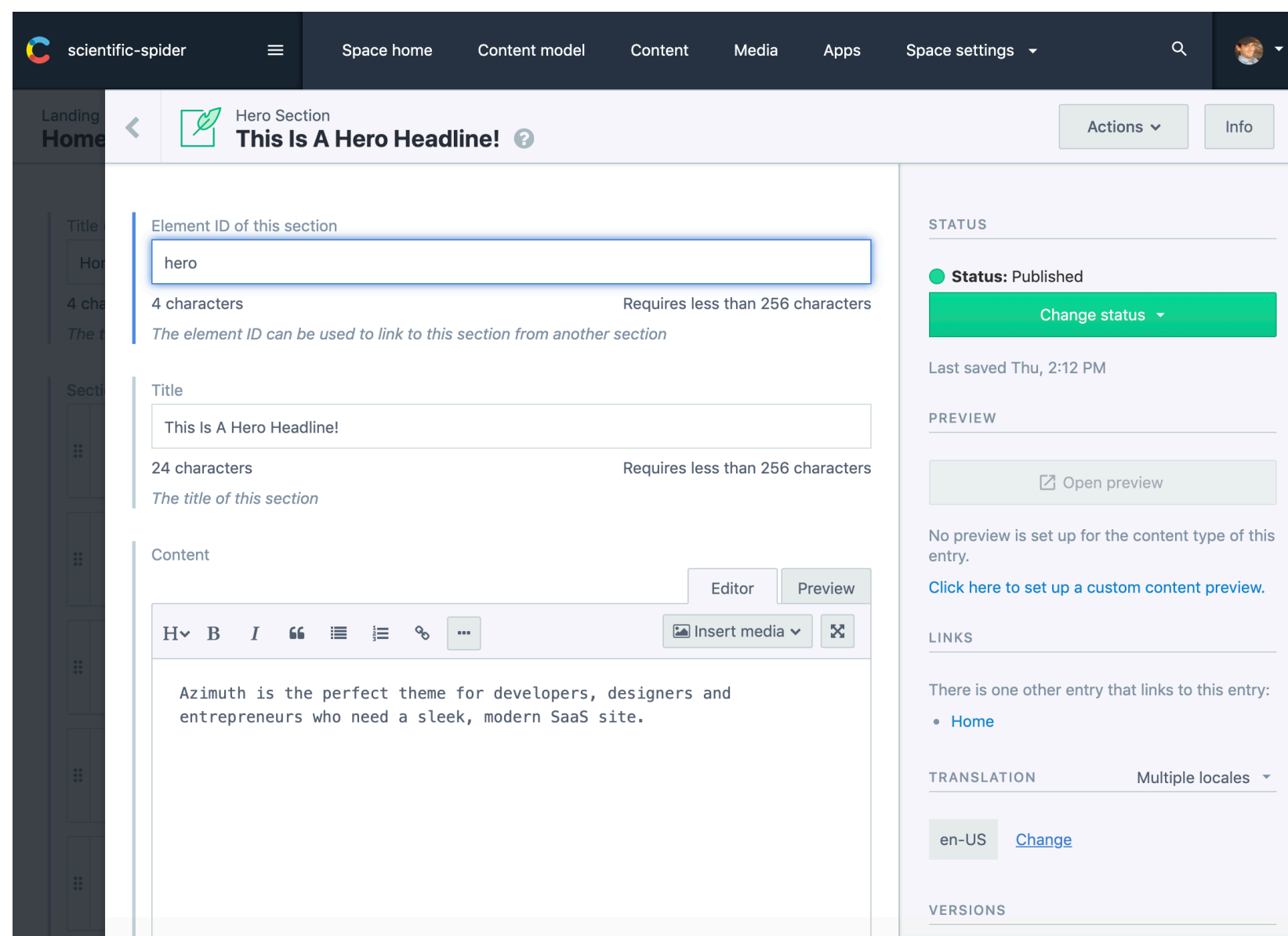
Chris Coyier released a post just yesterday that tries to organize different CMS into a chart based on some of the features to make it easier to understand how headless fits in.



'Source: <https://headlesscms.org/>

There's also headlesscms.org, which is maintained by Netlify, that lists all of the headless CMS with filters for supported static site generator, license and more.

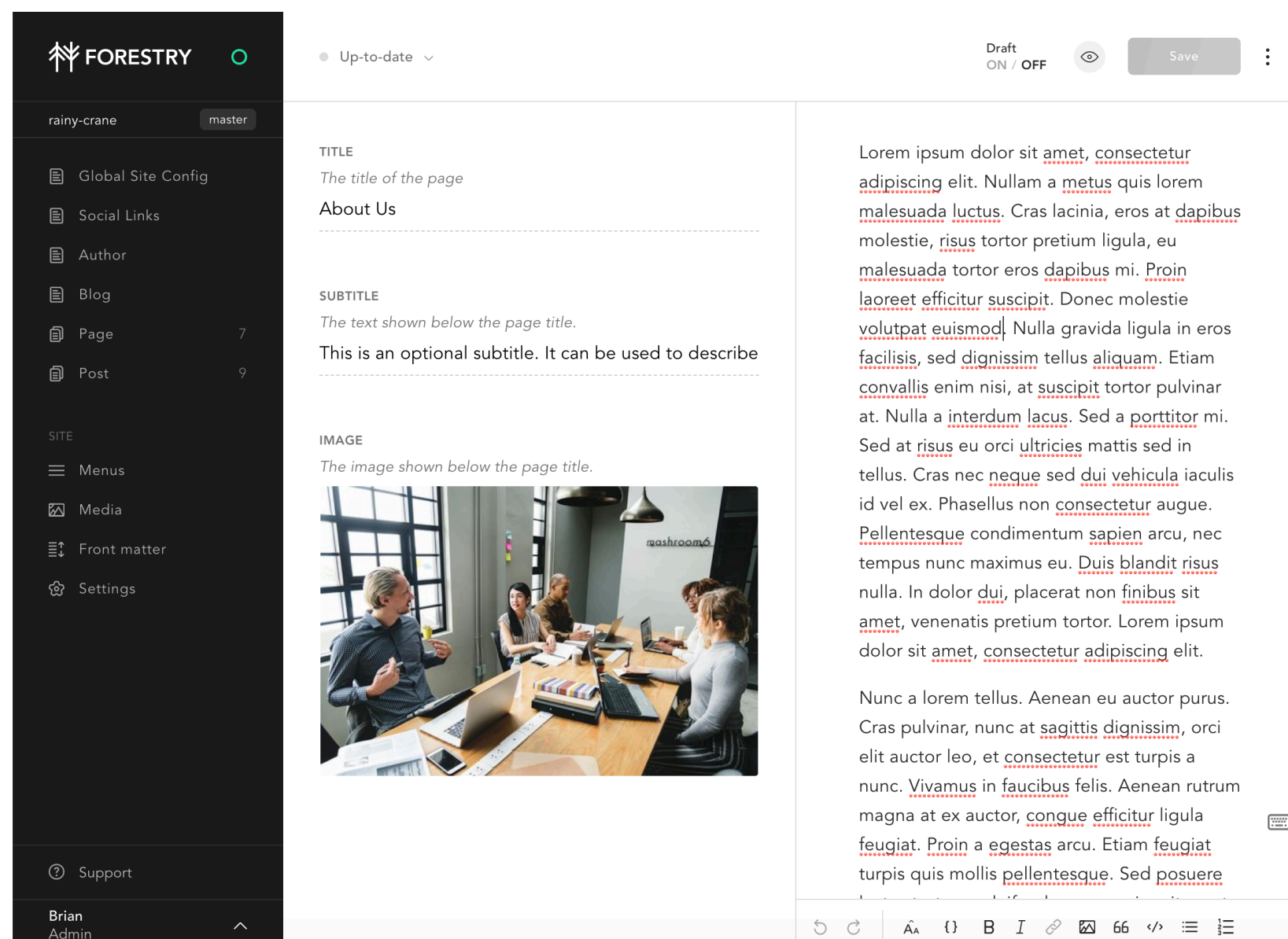
# Examples - Contentful



Contentful is probably one of the most well-known headless CMS solutions and perhaps also one of the most widely used. It is a more traditional headless-CMS in that all the content is stored in within Contentful and then accessed via an API. Let's look at the Contentful backend so you get a sense of what it feels like.



# Examples - Forestry



Forestry is an example of a git-based headless CMS. This means that Forestry stores content in a Git repository you create, but Forestry provides the tools to manage that content with an editing UI that understanding things like Markdown, YAML and JSON.

# Connecting Your JAMStack site to a Headless CMS

So now you have a CMS backend, but how do you integrate it with your front end site?

# Prebuilt Integrations

Contentful maintains pre-built integrations for:

- \* Jekyll
- \* Gatsby
- \* Middleman
- \* Metalsmith

Note that I tried to test the Jekyll gem. Installation and use is pretty simple, but it appears to not yet be compatible with Jekyll 4.0.0 which released relatively recently.

# Prebuilt Integrations

Forestry includes prebuilt starters for:

- Gatsby
- Hugo
- Gridsome
- Eleventy
- VuePress

As Forestry is a git-based CMS that doesn't actually store the content itself, rather than provide plugins, they provide starter projects.



# Use the API

Content can be brought in manually if you don't use one of the static site generators your CMS supports or you use a custom solution, provided it has an API. Some static site generators support bringing content from an API out of the box. You can also write something that locally calls the API and pulls content converting it to files or you can use something like a Netlify function that is triggered on deploy-building

# Stackbit

STACKBITBETA

THEME

Exto

SITE GENERATOR

Select a Site Generator

CMS

Select a CMS

DEPLOYMENT

GitHub + Netlify

My Project

Hi, I'm STACKBIT EXTO PORTFOLIO Theme.

The content will update in real-time. The recommended layout is used to show examples, but you can change it at any time.

Recent Work

Exto

Theme by Stackbit

A portfolio theme with a blog

Live Preview

This is a Big Hero Headline

Features

Azimuth

Theme by Stackbit

A sleek SaaS theme

Live Preview

Supports an on-site magazine. Add content to pages, blocks, collections, content, design, content and marketing.

Recent Work

Basic Rules For Walking In The Mountains

CONTINUE TO SITE GENERATOR

brian.rinaldi@gmail.com

# We Can Still Do Better



**Steve Polito Design**

@stevepolitodsgn



Replying to [@ShopTalkShow](#) and [@jamstackconf](#)

I so badly want to move towards JAMStack, but my clients really like "drag and drop" page layouts. Is there a headless CMS that can help?

5:11 PM · Oct 17, 2019 · [Twitter Web App](#)

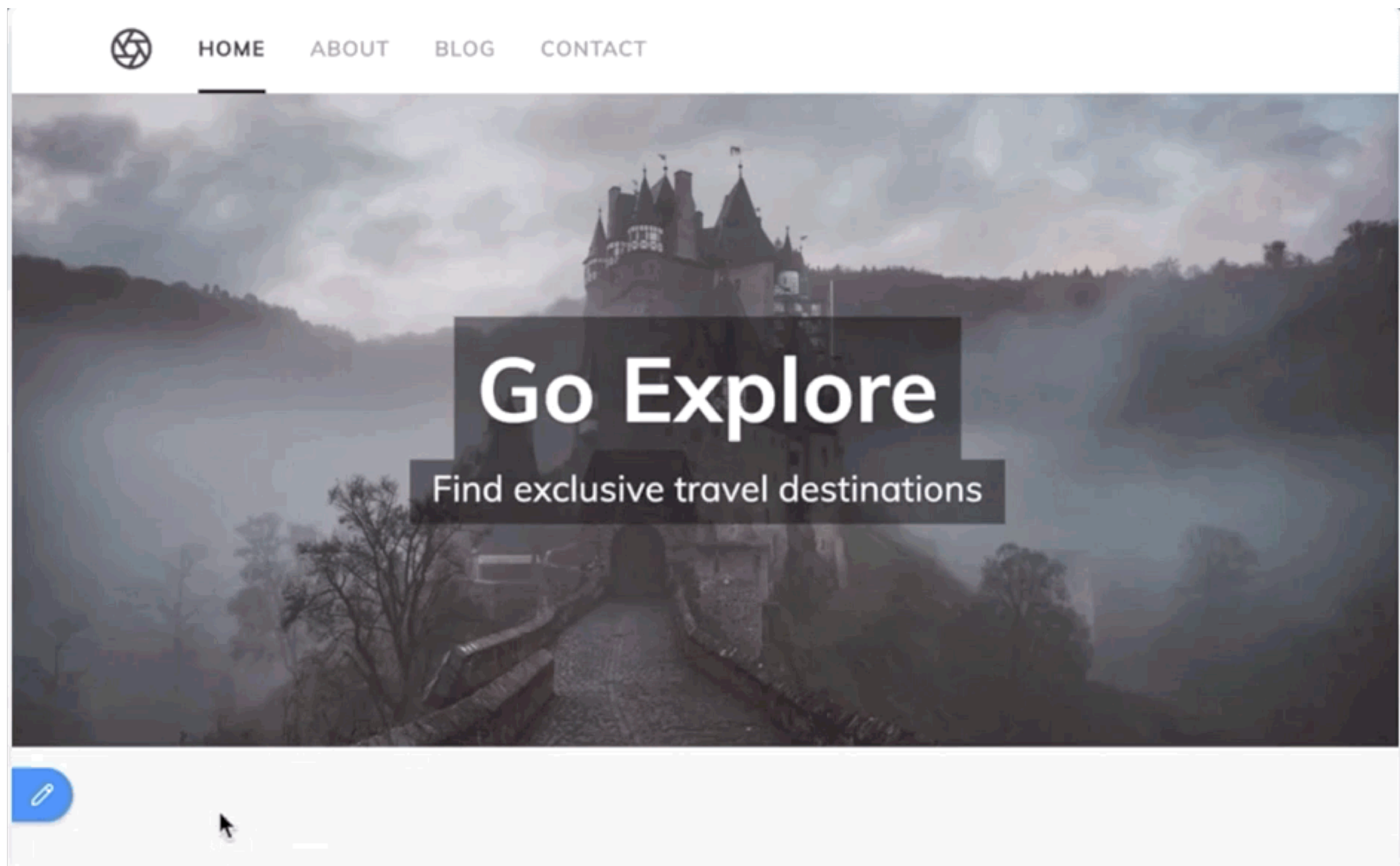
While I hope you agree that the editing experience has definitely improved, we're still not quite to the user experience many content editors have been used to.



# In the Year 2000

While we're at a point where JAMStack + headless is a real alternative to a traditional CMS, but it could be better. The administration of the site is still fully decoupled from the front end making some edits tough. Let's take a quick look at some ideas for how this can be improved.





TinaCMS is an open source inline editing experience built by Forestry. As you can see, it allows

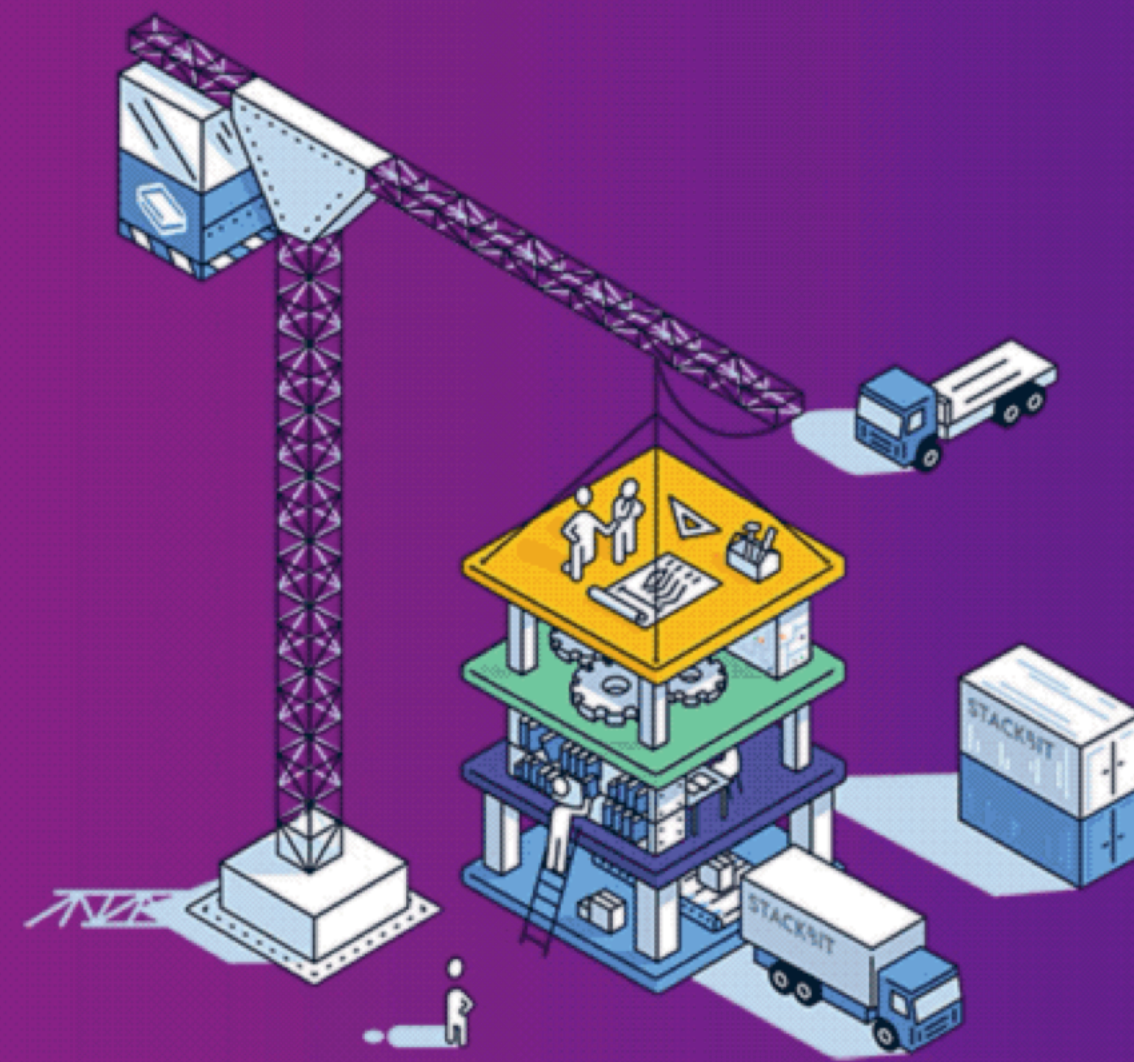




# This Is A Big Hero Headline!

Azimuth is the perfect theme for developers, designers and entrepreneurs who need a sleek, modern SaaS site

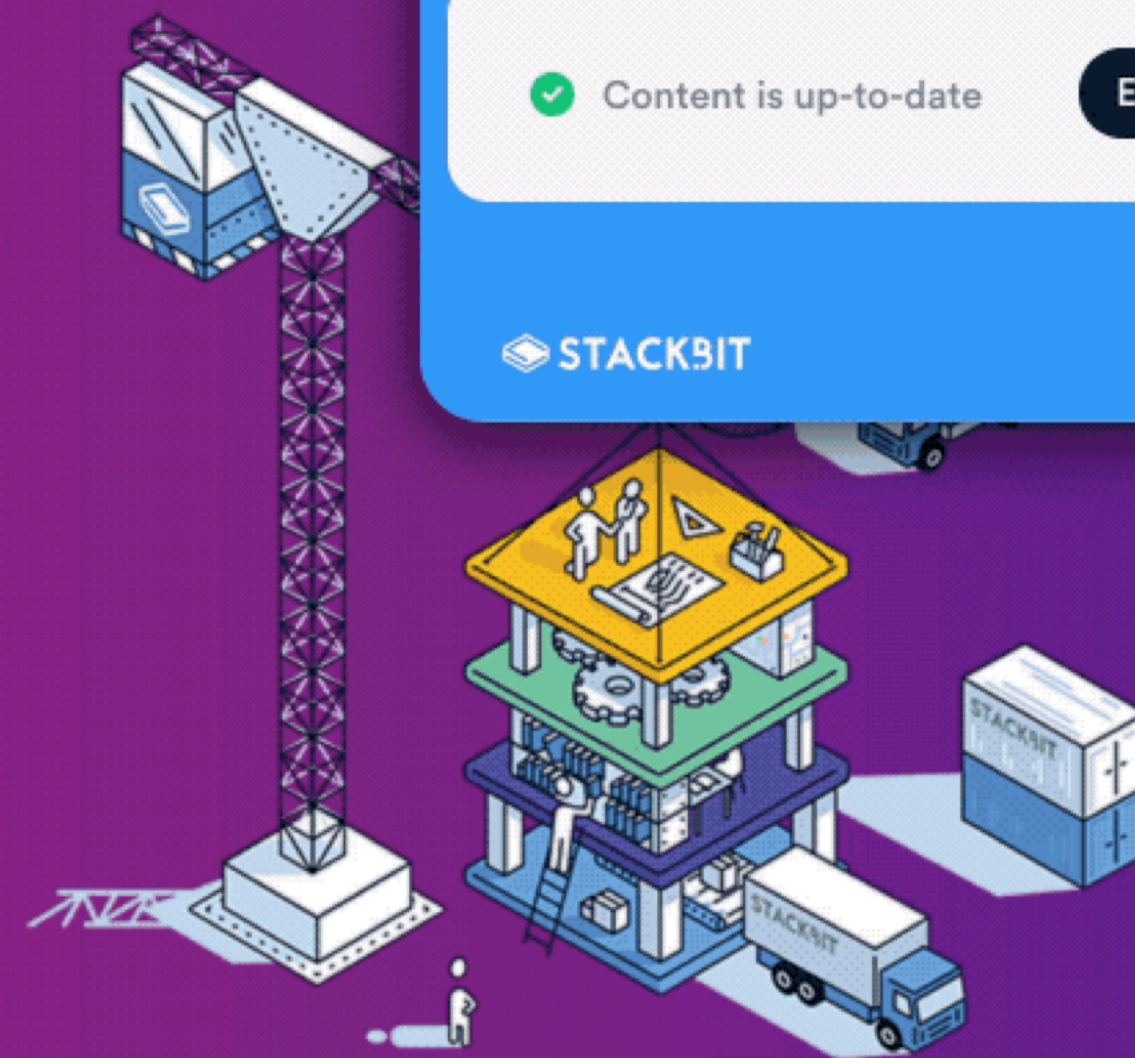
[Learn More Now!!!](#)





# This Is A Big Hero Headline!

Azimuth is the perfect theme for developers, designers and entrepreneurs who need a sleek, modern SaaS site

[Learn More Now!!!](#)

## Live Site

Last updated today at 3:18 PM

✓ Content is up-to-date

[Edit Site](#)

 STACKBIT

[stackbit.com/live](https://stackbit.com/live)



# Thanks!

Twitter: @remotesynth

Email: [brian@stackbit.com](mailto:brian@stackbit.com)