Twitter Thread by Jason Lengstorf

Jason Lengstorf
@jlengstorf

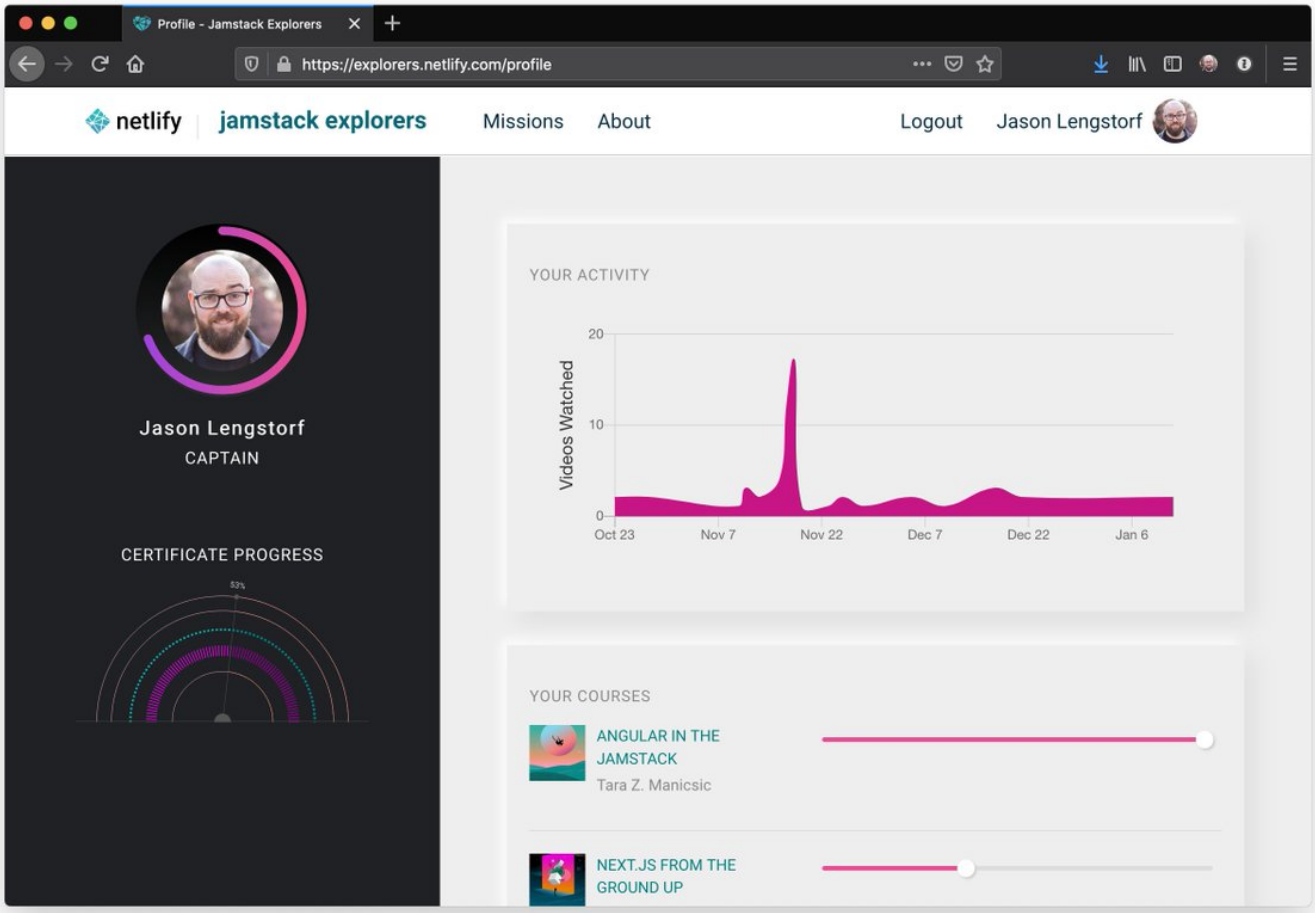
■ I'm a frontend dev. I work mainly w/HTML, CSS, & JS

last year I built stuff that doesn't sound like frontend work:

- custom APIs and DBs (both GraphQL & REST)
- user dashboards
- video manipulation

but it all *felt* comfortable & within my skillset

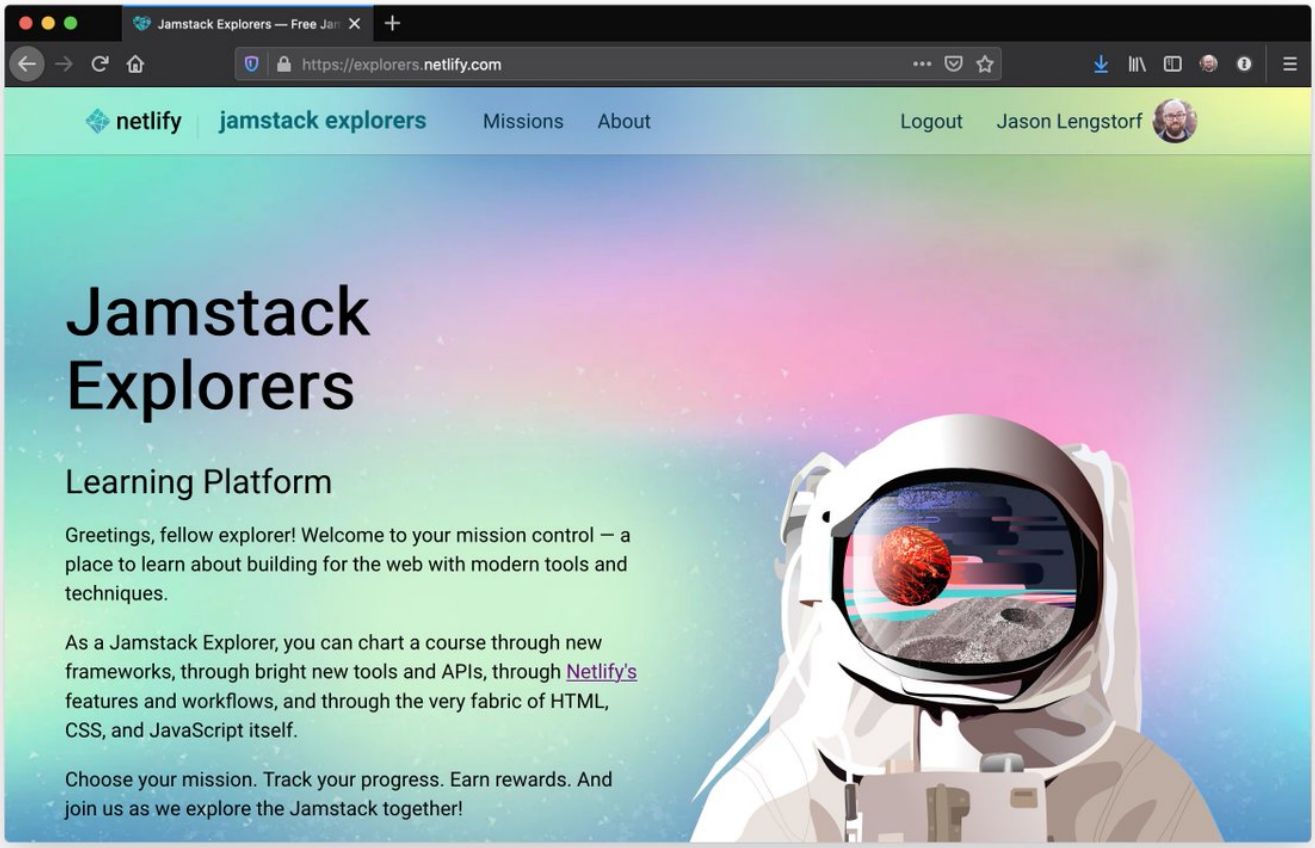
how? a thread: ■



I do all sorts of demos and "hello world" projects, but I want to focus on real-world apps only in this thread. specifically, I'll talk about this production project:

■ Jamstack Explorers ■■■

this is a load-bearing app that a lot of people depend on



to build Jamstack Explorers, we needed:

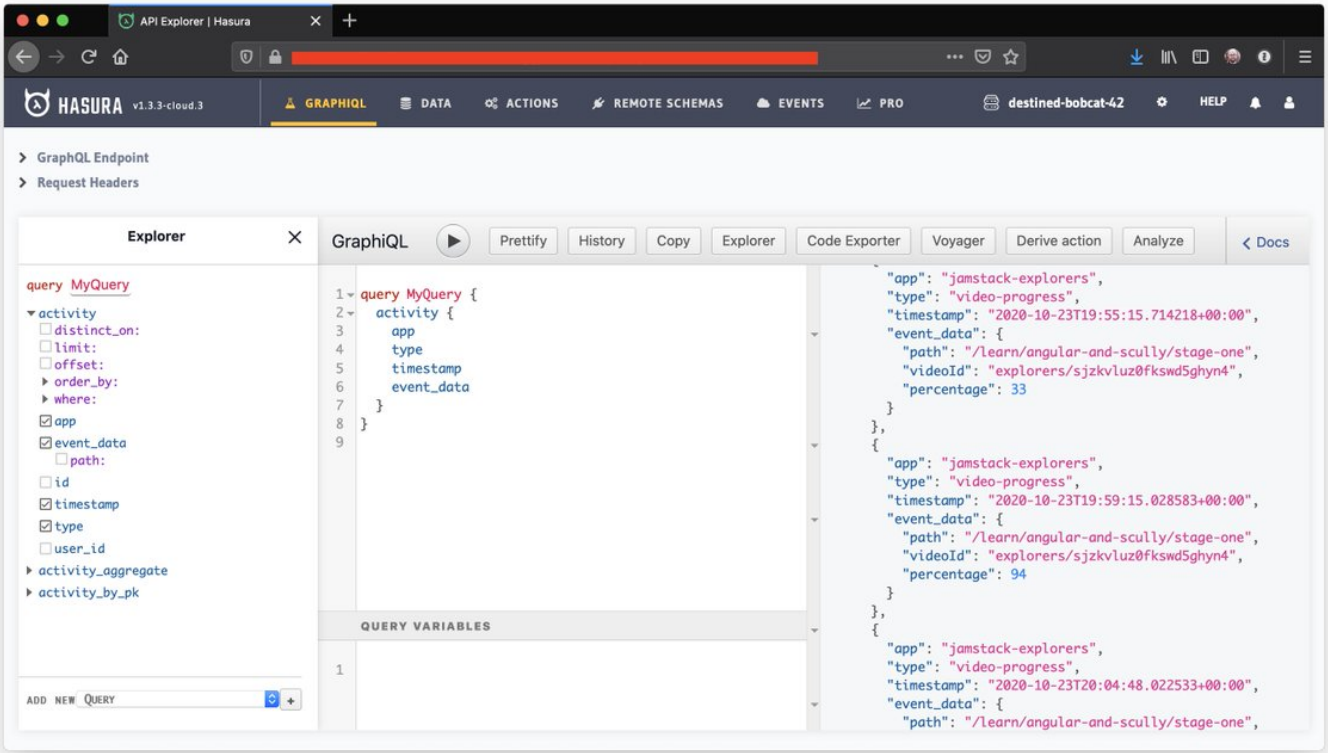
■ a custom database to track mission progress
■ a content management system
■ user authentication
■ video manipulation

we were a small team of frontend devs and we needed to ship quickly — this was a daunting todo list ■

■ custom database

using @HasuraHQ Cloud, we were able to create a new DB, configure it, and test the API all from a web interface — no config files or server setup required

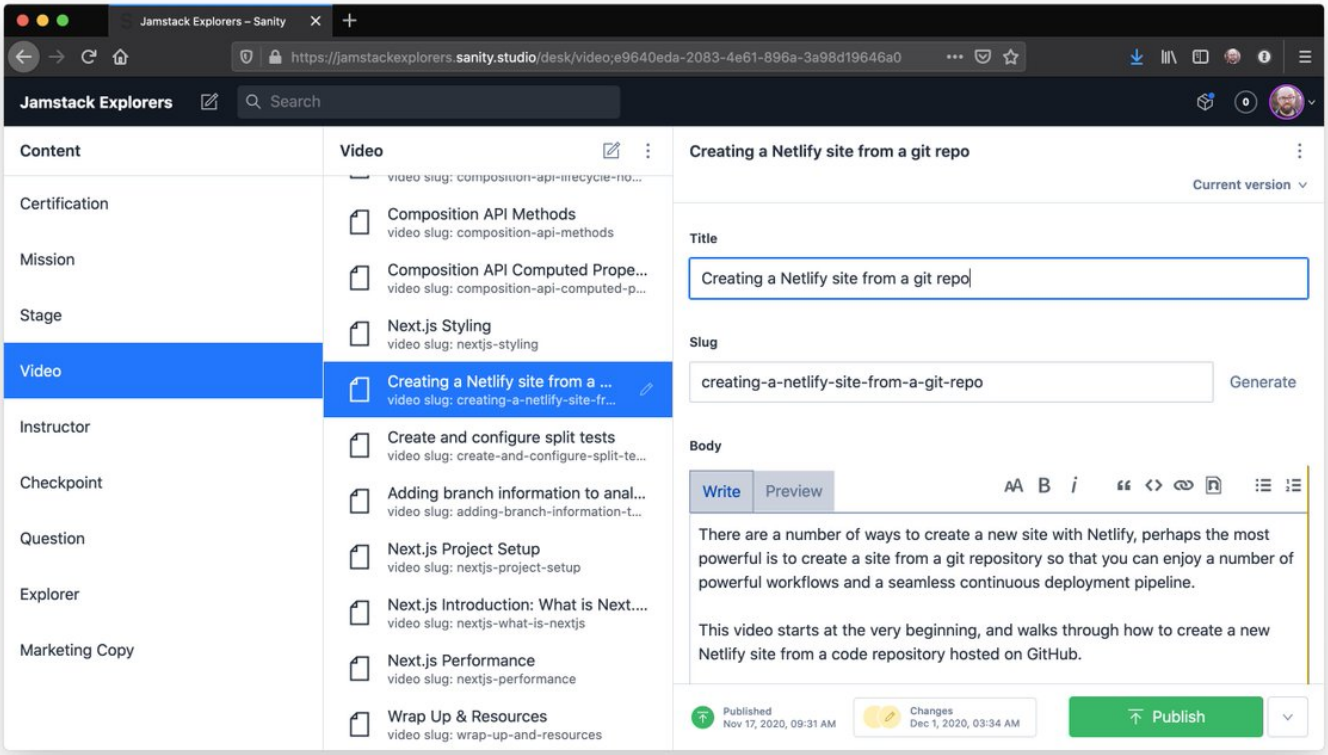
we got it running in a day & I never felt like I was in over my head, even though DBs make me nervous!



■ content management

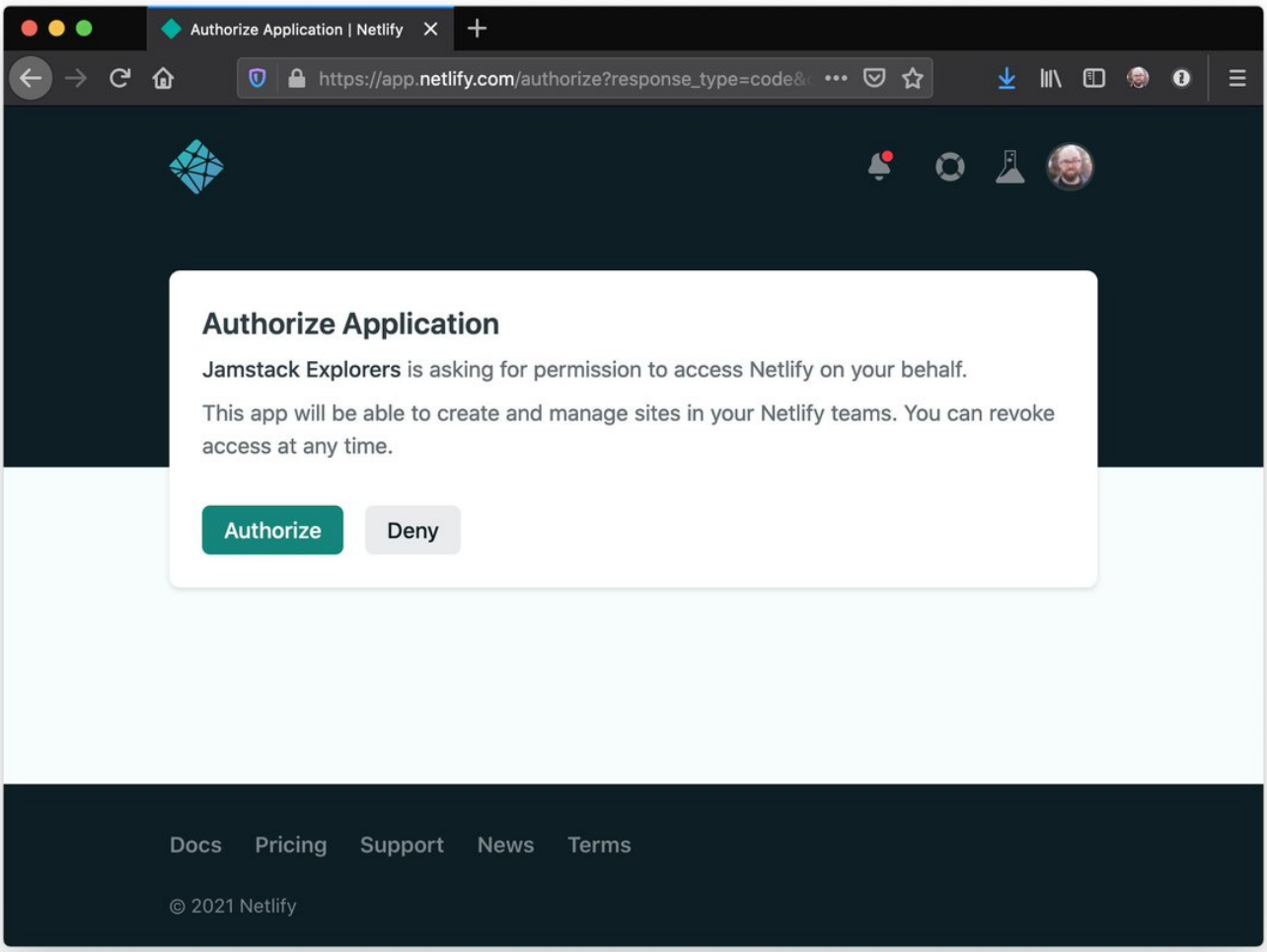
we chose @sanity_io as a CMS. we used their CLI + JSON schema to set it up for Markdown + @mdx_js, exposed through a GraphQL API

the docs were great — it still felt like I was well within my frontend wheelhouse while we set this up



■ user authentication

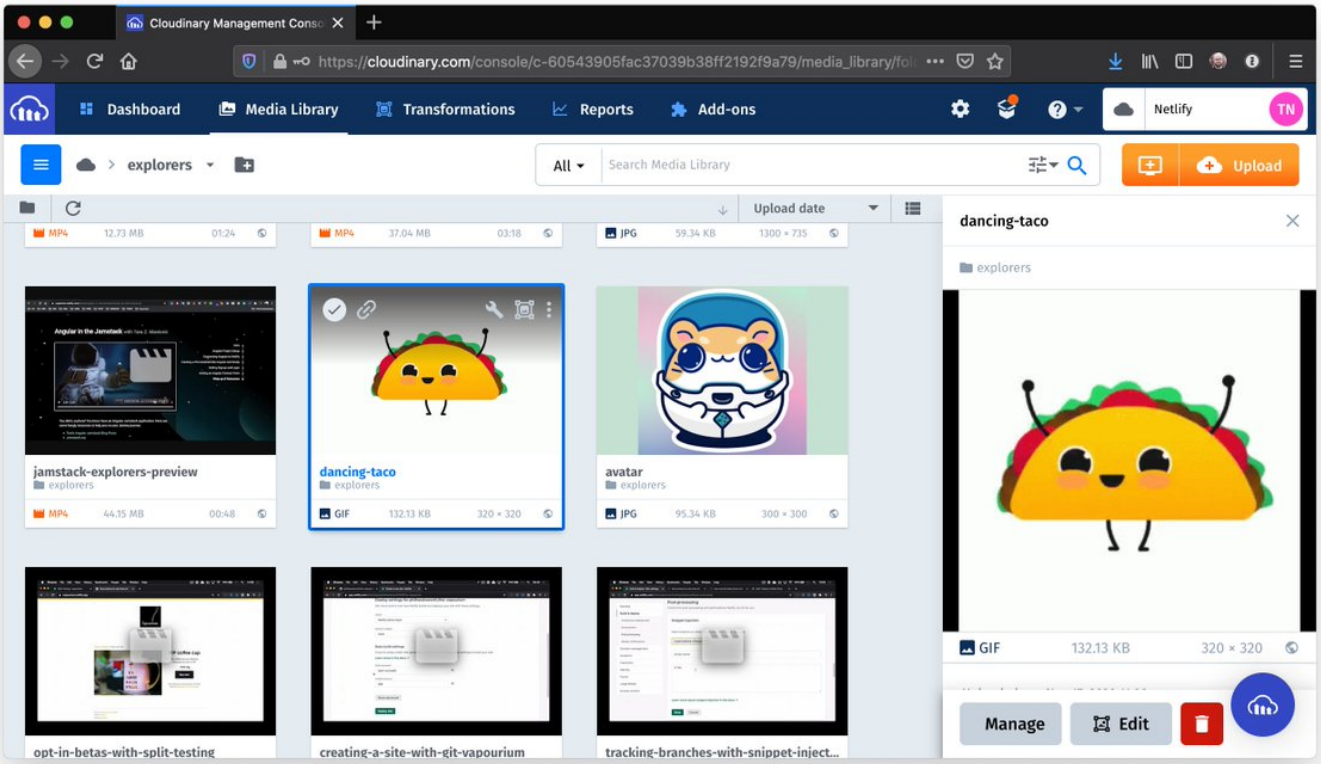
OAuth2 is a challenge to set up, *BUT* by using @Netlify Functions, we only had to figure out how to handle auth — not how to set up/deploy a server, listen for requests, AND handle auth. it was head-bendy, but it's still written in JS, so it felt familiar



■ video manipulation

this scared me — like, where do we even start?

fortunately, @cloudinary made it super approachable! we upload videos through Cloudinary's UI, then use the URL-based API to handle transformations like auto-generated title cards & auto-inserted bumper videos



■ remember: we did all of this — user dashboards, custom databases, on-the-fly video editing — in a Next site that deploys to Netlify with a few serverless functions and SaaS tools. we deployed without ever having to think about containers or kubernetes or SSH-ing into a server

it felt GREAT to be able to build all of that functionality without having to step very far outside my primary skillset. I got to be a frontend developer, and when we needed more, we stitched in third-party services and relied on serverless to keep things approachable ■

this approach also made us SO MUCH FASTER. we built all of that functionality as a team of frontend devs in a couple months while *also* working on a pile of other projects *and* making all the video content for it ■

what we *didn't* do is go into crunch mode to ship ■

while I'd love to say it's because my team is incredible (they are), the truth is that the Jamstack architecture with SaaS powering backend needs will make teams faster no matter who they are. there's less context switching, fewer layers to navigate, and clearer system boundaries

➡■ my major point here is: I am *so stoked* I get to be a frontend dev right now. I never would have dreamed I could build even half of what I helped build

I'm even *more* stoked to see what the devs in this incredible community can do with all these capabilities! ■