Using Node.js and React with Drupal

Eric Hestenes
George Lucas Educational Foundation
Hello!

Eric Hestenes  @ericsfeed
Director of Engineering

- AI/human factors
- Financial services technology
- Helping out education foundations and schools
- Peer to Patent – open government; OSTP; dept of treasury
- Drupal consulting
- Agile software development → delivering useful software
edutopia

• Nonprofit; small team; website: www.edutopia.org
• We shine a light on what is working in education

Our website is a single page application with a Drupal back end:

Front End App
Node, React

Data Pipeline

Data Services
JSON API
Drupal 8
Our Journey

- Late 2016 we decided to improve our platform
- Launched a new rebuilt site in January 2018
- We had many challenges, and overcame most.
- At conferences, did not find examples like ours

The goal today:

- To share some of what we learned
- Patterns may be interesting or helpful to others
Challenges we faced:

- Initial decision
- How to get data from the Drupal API
- Resolving URLs
- Previews
- Media Handling
- Performance
What path would you choose?

- We had to choose a path
- Every upgrade path forward implied change and learning
- If we go with Drupal 8 + javascript - **What does that really mean?**
- This combination is kind of new – **Does it actually work?**
Why use Drupal as a service?

✓ Already a Drupal site and have lots of legacy data

  Do not want a proprietary CMS
  Needs include typical content management use cases
  Want a solid baseline that can evolve over time
  Keep content complexity out of the app, into the CMS

✓ Wanted to go deeper into the front-end universe

  Tap into front-end software, community, and ecosystem
  Wanted features that are not in Drupal
Is using Drupal this way supported?

- Community focus has been building up over time:
  - API-first initiative
  - JSON API and GraphQL support has improved over time.
  - Solid core of content management and editing capability
  - Wonderful open source community
- Contributor on-ramps like Contenta, Gatsby
- Drupal hosting ecosystem: can spin up Drupal as a service.

*Supports for API consumers have improved over the last 24 months.*
Simple setup.

$ sudo composer require drupal/jsonapi

→ http://example.com/jsonapi/node/article

You can also try this out:

$ sudo composer require drupal/graphql

https://www.drupal.org/docs/8/modules/json-api/fetching-resources-get
How to get the data out of the back-end

• JSONAPI or GraphQL?

• Use JSON API with sparse fieldsets

  Out of the box JSON API is verbose; get only the fields you need

• Compress and proxy API responses.

  E.g. a CDN, Node + Varnish or Fastly.

• pm2 clustering allows multi-core workers; avoid blocking

• We needed a url resolver on the server side. What’s that?
You might want a resolver

• Due to site evolution, legacy Drupal pages may not be strict in handling URL routes.
  
  E.g. /blog/ is now /article/

• Routes are content!

• The front-end app does not know which endpoint to hit.

• We needed a more robust URL handling solution

• Redirect handling is also in play
How we resolve URLs

• We created a custom endpoint that uses a parameter called ?url-slug
• The resolver determines the proper react component type for a given URL based on the Drupal content type.
• We use JSONAPI to serialize the object, and return that.
• This allows us to stage changes to URL patterns, provides backward support for legacy URL patterns.
• We take the URL and resolve it through Drupal, and pass the correct data back to the API
Previews
Another Challenge: Previews

- We’re decoupled – that means that Drupal cannot render pages. So our editors could not preview content.
- Also JSONAPI does not directly support revisions
- We needed a way to get a specific revision and then configure the front-end to use that revision
Edit Article Creativity Can Unlock Resilience in Educators

Title *
Creativity Can Unlock Resilience in Educators

Hide Intro Field
If checked, the content in the Intro/Dek will not be shown on this node's page.

Intro (dek)

When we are creative, we are resourceful and able to problem-solve in new and original ways.

Summary *
When we are creative, we are resourceful and able to problem-solve in new and original ways.

Draft

Last saved: 08/10/2018 - 14:21
Author: Rebecca Alber

Create new revision
Revisions are required.

SCHEDULE PUBLISHING
Add new Publish Date

Revision log message
Briefly describe the changes you have made.
Uses the Workflows module (in core) to allow saving a draft revision.
Previews: The details

- Modify Drupal so the revision id is baked into editor links. Then we use that to fetch the revision of the entity and generate the API response.

- On the node VM, we have a separate docroot for the preview, which allows managing authentication as config and we can lock down preview content.

- The same engine we use to render the main site also renders previews.

- Override a normalizer and JSONAPI so we can access an extra parameter that contains the revision id. Return the revision instead of the published version. (jsonapi v1)

https://github.com/edutopia/preview-example
Flexible Landing Pages

- Customize and tailor landing pages
- Allow editors to create more complex pages
- With no front-end code deployment
- Keep complexity out of the front-end

Can you do this in a decoupled environment?
Layout examples

BRAIN-BASED LEARNING
What Is the Social Brain?
The research is in: Learning is a social endeavor.

INTEGRATED STUDIES
Using Science to Bring Literature to Life
Combining science and literature can help students engage more deeply with both subjects.
Edward Kang

INTEGRATED STUDIES
Using Stories to Teach Math
Teaching literacy and math at once helps make the most of class time while deepening young students’ understanding in both subjects.
L.L. Barkat

INTEGRATED STUDIES
Sparking Curiosity and Shining Real-World Problems
See how students apply their literacy skills to explore science problems outside classroom walls.

TRADE PLACES
Birmingham Covington: Building a Student-Centered School
Educators take on the role of guides and motivate students to direct their own learning.
Holly Korbey

BRAIN-BASED LEARNING
The Social Classroom
Increasingly, modern classrooms support group work and peer-to-peer collaboration. The science says that’s right on.

BRAIN-BASED LEARNING
Masters of Social Learning
We’re hardwired from birth to decode facial expression, posture, and tone of voice—and to work collaboratively.

BRAIN-BASED LEARNING
When the Social Brain Misfires
Sometimes our powerful urge to belong can hurt us. We look at how stereotypes undermine learning.

LITERACY
Project: Every Kid Has a Story to Tell
A video project helps students hone their multimedia storytelling skills, while developing greater self-awareness and empathy for their peers.
Mindy Ahrens

TECHNOLOGY INTEGRATION
Reluctant Adopters and Technology Initiatives
Addressing the valid concerns teachers have about new tech tools helps them accept change and builds positive relationships.
Theresa Cullen

FLIPPED CLASSROOM
Making Your Flipped Classroom More Human
A former high school biology teacher shares what he learned from his mistakes in flipping his classroom instruction.
Michael Ralph

SOCIAL AND EMOTIONAL LEARNING
Project: Every Kid Has a Story to Tell
I focus on arts and sciences to foster skills that spare them for college and careers.

SOCIAL AND EMOTIONAL LEARNING
Important Questions to Ask Your Students

LITERACY
Double-Duty Writing Instruction

https://www.edutopia.org/topics
Editing layouts

A composite page is made from component blocks that can be serialized as JSON.

Editors can choose from a set of layout types for each block.

Editors can change the ordering of components and customize various parameters such as filters using taxonomy.
The Media Challenge

• We had a master image library in Drupal 7

  • Stand-alone Drupal

  • Automates cropping a single master image at various aspect ratios

  • Support for focal points

• Needed a way to generate responsive images

• A way to use those images in our content

• Did not want to expose files on our backend server
Responsive Images – How it works

- Master image library in Drupal, which uploads master images to S3
- Cloudfront CDN redirects to Lambda if no derivative is found
- Lambda generates image derivatives on the fly; stores them in S3
- Uses Drupal to define the responsive image sizes and breakpoints
- Drupal has the config but the work is happening in Lambda.

Learn more: https://aws.amazon.com/answers/web-applications/serverless-image-handler/
The Upside of Decoupling

- Get to a better *separation* of concerns
- Allow higher degree of *control* over the front end & middle tier
- Tap into this Node and Javascript developer community and *ecosystem*
- The *velocity* of front-end changes is much higher than the backend.
- The backend can be more *stable*.
- We have a much smaller footprint of Drupal modules
- We can remember activity in the session and are *smarter* about network usage.
Performance tuning

**Use a performance budget.** Our was: 4s on mobile 3G, sub-second stretch goal

- Pay attention to the bundle size.
- Limit trips to the network
- Use code splitting to load just the code in use
- Implement CSS in JS, such as emotion.js (sass, aphrodite, then emotion)
- Leverage progressive web apps (service worker) and Lighthouse
  
  https://developers.google.com/web/progressive-web-apps/checklist

*Performance was the tie breaker in many key decisions*
Creating the perception of speed

- Storing API results in local state.
- Pre-fetching API calls.
- Lazy loading of images and content
- Fetch on hover
- Relentless monitoring and audits: webpagetest.org, lighthouse, devtools, pagespeed insights

You can change perception from seconds to sub-seconds
Gotchas

• For service workers, look at Google WorkBox.
  https://developers.google.com/web/tools/workbox/

• Do not cache sw.js

• By making the app too smart, you can reveal less to Google and lose SEO

• Server-side rendering is required for SEO and social media

• Google AMP support required another node application.

• Beware webfonts
  
  Self hosted > good webfont implementation > bad webfont implementation
Our fear: the Framework learning iceberg

- What functionality goes in the javascript stack?
- Getting to know: express, react router, redux, webpack
- Explosion of rapidly evolving packages and choices
- PWAs and those scary service workers
- New tool chain, linting, prettier, test automation, environments
- Integrations like enewsletter, site search, social media, analytics

Coming up to speed takes time and requires experienced help. Lean hard into established patterns.
This is where we ended up.
What to expect from decoupled Drupal?

+++ a workable JSON API service
+++ solid, predictable, fast API performance
+++ fits into contemporary tool chains – git, composer
++ support for implementing a preview system
+ a resolver (urls matter to SEO, leverage pathauto)
+ vendor support for spinning up and hosting Drupal

Need: better onboarding recipes for javascript developers
Final Thoughts

• Yes you can leverage Node and React best practices together with Drupal

• JSON API works and it is fast. But the implementation is changing as it moves into Drupal core – coming soon. Check out GraphQL too.

• For Drupalers, Pandora’s box is open to many new user experiences.

• Decoupled Drupal is scaled down Drupal, with features removed. This reduces the maintenance footprint, and right-sizes scope of the service. In this case, less is more, less is good.
THANK YOU

Any Questions?
Eric.Hestenes@Edutopia.org / @ericsfeed
Credits

Edutopia
Tyler Burke
Nichole Davison
Pamela Hastings
Eric Hestenes
Stephen Merrill
Rob Wohleb
and more…

Lullabot
Greg Dunlap
Jeff Eaton
John Hannah
Darren Peterson
Matt Robison
Carwin Young
Sally Young