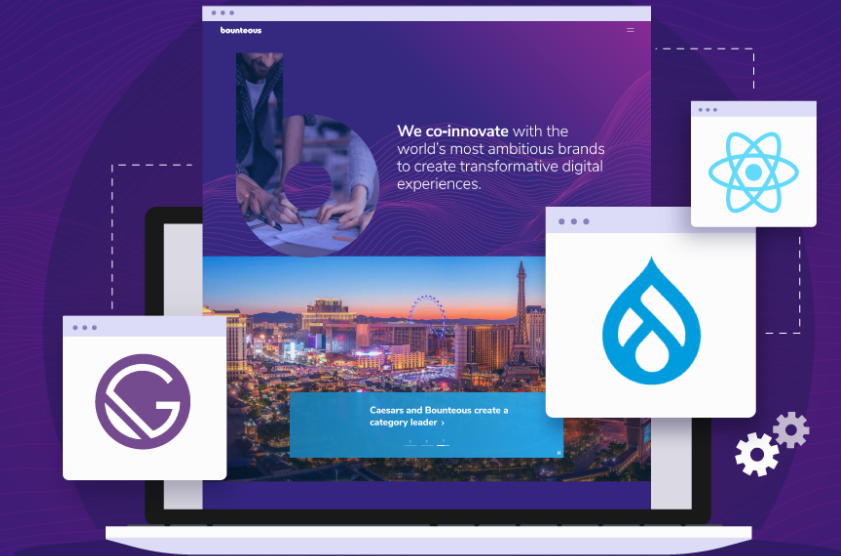


Iteratively Decoupling Your Existing Drupal Site With Gatsby

DrupalCon North America
April 13, 2021

Resources: <http://bit.ly/iterative-dc>

- Brian Perry
- Matthew Ramir



YOUR HOSTS

A Couple of Decouplers

bounteous



Brian Perry

Front of the Front End

- Lover of all things components...
- ...and all things Nintendo (currently playing Ori and the Blind Forest)
- @bricomedu
- brianperry.dev



Matt Ramir

Back of the Front End

- Lover of all things devops
- ... and all things outdoors (excited its hiking season)
- @crasx
- Ramir.dev

A photograph of two men sitting at a table, working on laptops. The man on the right is smiling and looking at his laptop. The man on the left is wearing glasses and looking at his laptop. A can of beverage is visible on the table. The word "bounteous" is overlaid in white text in the center of the image.

bounteous



Justifying a Decoupled Build

Struggling with cost vs benefit

Front End Structure

A single codebase serving multiple front ends

Hydrating Gatsby

How we got data from Drupal to the frontend

Workflow Changes

The impact of introducing static builds

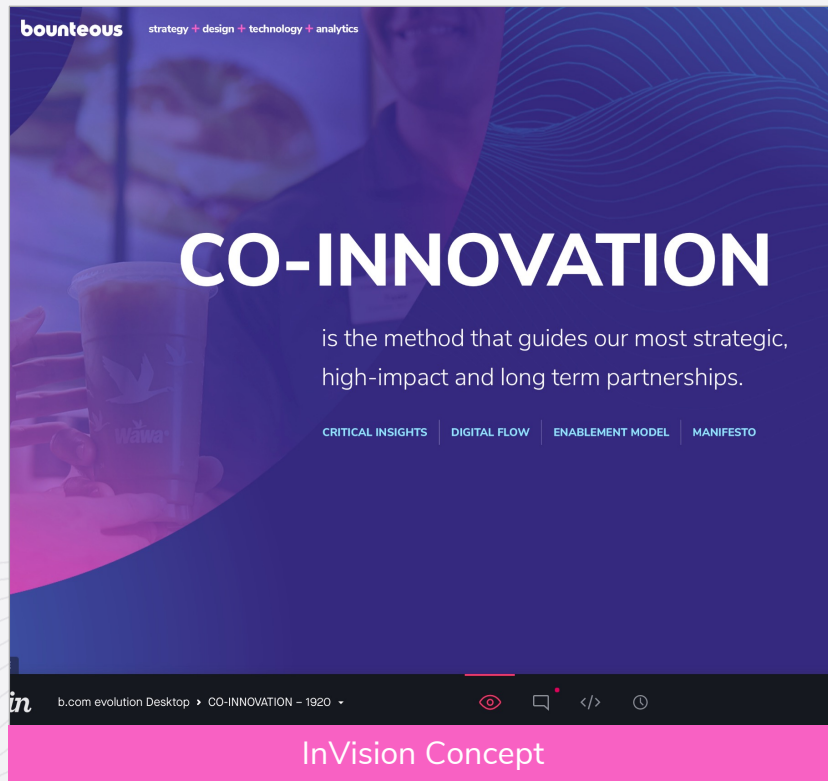
The Blurring Line Between Front Ends

The impact of managing multiple component libraries

The Ask

A phased approach to a redesign

- Two templates with a completely new design
- Large amount of new behaviors and animations not present on the existing site
- Rest of site mainly gets a cosmetic overhaul, updating global styles in line with new design



I've seen this movie before

We've considered decoupling in the past

- Unable to justify full scale overhaul
- Needed to balance effort with competing priorities
- Some content on site might not benefit greatly from decoupling

Redesign concepts put us at a similar crossroads

- Leveraging a JS framework was desirable for new design concepts...
- ...but not necessary for the majority of the site
- Decoupling entire site would greatly delay what is essentially just shipping 2 new pages

So what changed this time?



Could we start by only decoupling two pages on our existing site?

- Us, not sure if this was even possible

Turns Out We Could 🎉

Initially launched with:

- Two pages built statically with Gatsby
- All other content rendered by Drupal

Long-term groundwork:

- Any page on the site can be rendered primarily by either React or Twig



Justifying a Decoupled Build

Struggling with cost vs benefit

Front End Structure

A single codebase serving multiple front ends



Hydrating Gatsby

How we got data from Drupal to the frontend

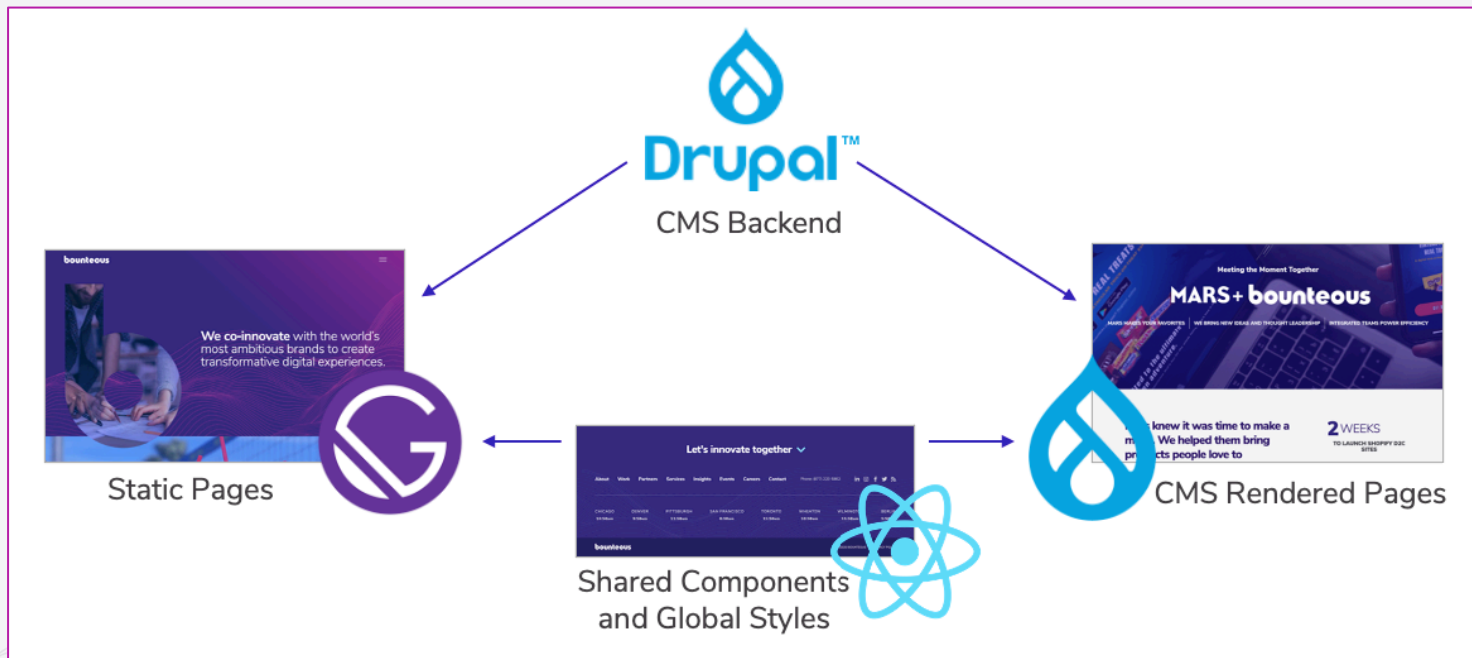
Workflow Changes

The impact of introducing static builds

The Blurring Line Between Front Ends

The impact of managing multiple component libraries

One Site, Multiple Front Ends

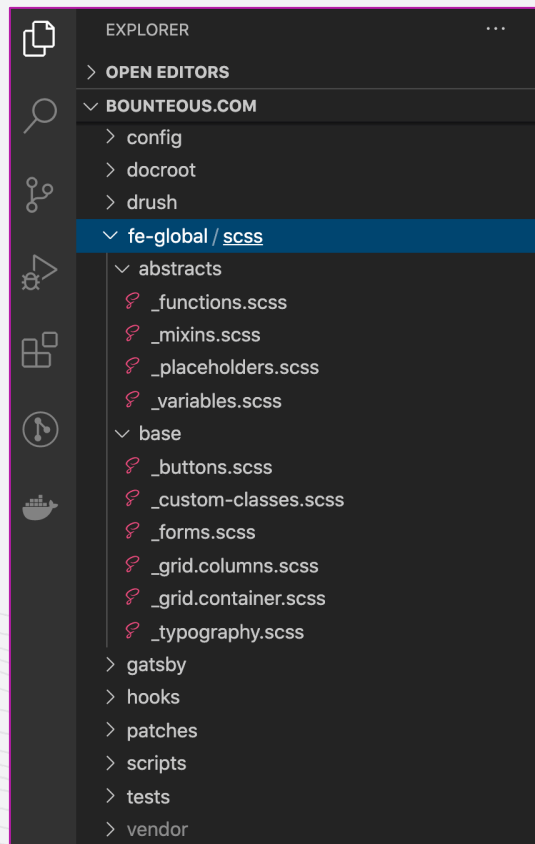


Project Structure

- Monorepo style approach in single repository
- Three main top level directories:
 - /fe-global
 - /docroot (drupal)
 - /gatsby

Front end globals:

- Sass partials
 - Global styles
 - design tokens



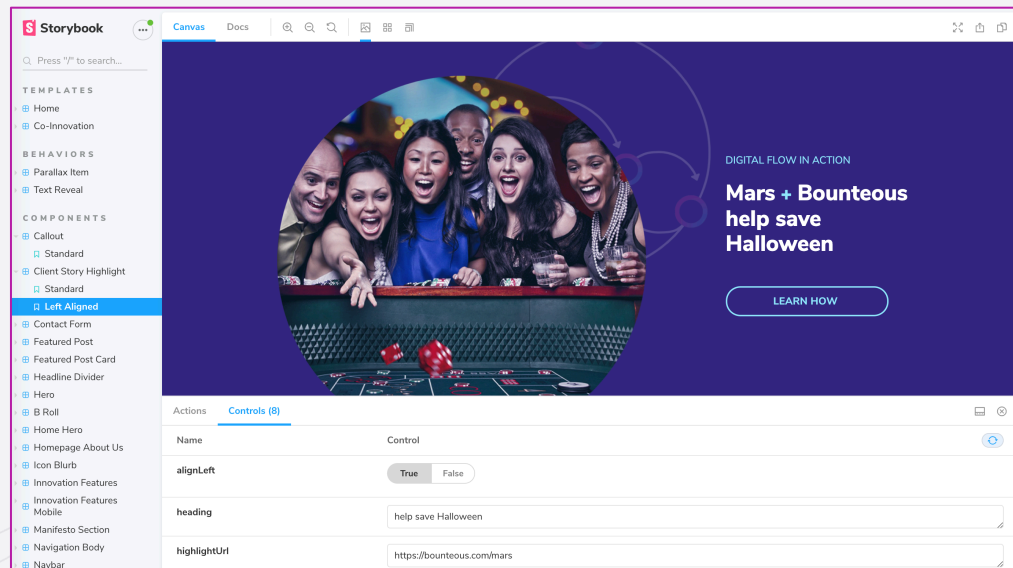
React

Focused on building functional components with as little internal state as possible

Different data contexts:

- Data sourced from Gatsby's GraphQL API
- Directly within Drupal
- Hardcoded data

Heavy use of storybook during initial phase



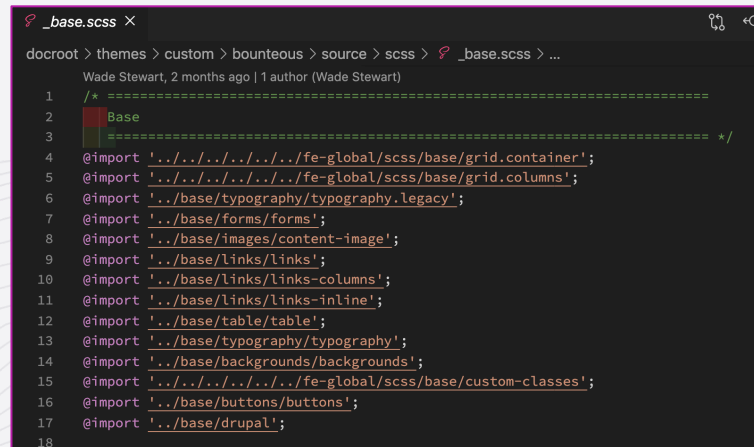
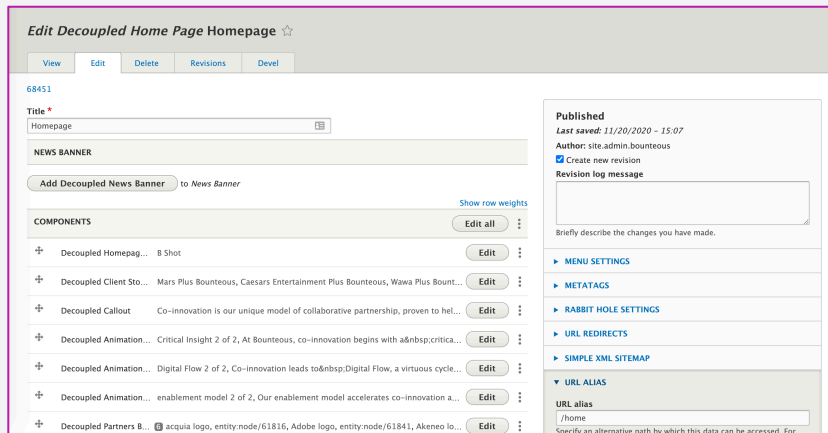
Drupal

Content Model:

- Created new content types for decoupled page templates
- Continued using paragraphs
 - Surprisingly reasonable to access data via GraphQL API
- High level of layout flexibility

Theme:

- Still have traditional Drupal theme
- Imports partials and tokens from fe-globals as needed



Justifying a Decoupled Build

Struggling with cost vs benefit

Front End Structure

A single codebase serving multiple front ends

Hydrating Gatsby

How we got data from Drupal to the frontend

Workflow Changes

The impact of introducing static builds

The Blurring Line Between Front Ends

The impact of managing multiple component libraries



Configuring gatsby-source-drupal

```
64   {
65     resolve: `gatsby-source-drupal`,
66     options: {
67       baseUrl: process.env.DRUPAL_BASE_URL,
68       basicAuth: {
69         username: process.env.AUTH_USERNAME,
70         password: process.env.AUTH_PASSWORD,
71       },
72       // In this section we prevent the system from pulling in ALL nodes of a given type.
73       disallowedLinkTypes: [
74         `self`,
75         `describedby`,
76         `file--file`,
77         `node--casestudy`,
78         `node--insights`,
79         `node--employee`,
80         `path_alias--path_alias`,
81       ],
82       // There are many cases where we want to include a specific node that was
83       //disallowed above, this forces the system to pull the related objects for nodes that have them.
84       filters: {
85         // Use includes so only the files associated with our decoupled content
86         // types are included.
87         "node--client_story":
88           "include=field_cs_client_logo,field_cs_tech_logo,field_cs_teaser_images",
89         "node--client_quote":
90           "include=field_client_logo,field_client_quote_pinned_pages",
91         // Decoupled Home Page
92         "paragraph--dhp_about_us_slides":
93           "include=field_dhpaus_bg_img,field_dhpaus_fg_img",
```

Wiring data to components

```
3 import { graphql } from "gatsby"
4 import HomeHero from "../../components/HomeHero/HomeHero"
5
6 const HomeHeroProvider = ({ paragraph }) => {
7   const fgImg = {
8     alt: paragraph.field_dhp_fg_img[0].alt,
9     src:
10       paragraph.relationships.field_dhp_fg_img[0]?.localFile?.childImageSharp
11         ?.fluid,
12   }
13
14   return (
15     <HomeHero delay={0.25} duration={0.75} bRoll={fgImg} imgContext="gatsby" />
16   )
17 }
18
19 export default HomeHeroProvider
20
21 export const HomeHeroFragment = graphql`
22   fragment HomeHeroFragment on paragraph_dhp_hero {
23     id
24     field_dhp_fg_img {
25       alt
26     }
27     relationships {
28       field_dhp_fg_img {
29         localFile {
30           id
31           childImageSharp {
32             fluid(maxWidth: 580) {
33               ...GatsbyImageSharpFluid_withWebp_noBase64
34             }
35           }
36         }
37       }
38     }
39   }
40 }
```


Wiring data to components

```
32   scroll_build_detail: ScrollBuildDetailProvider,  
33   decoupled_download_cta: DownloadCtaProvider,  
34   component_group: ComponentGroupProvider,  
35   decoupled_innovation_features: InnovationFeaturesProvider,  
36   decoupled_quote_card_no_headshot: TestimonialQuoteProvider,  
37 }  
38 // Drupal rendered  
39 drupalRenderedParagraphs.forEach(value => {  
40   componentMap[value] = DrupalRenderProvider  
41 })  
42  
43 const paragraphsToComponents = paragraphs => {  
44   // Create a new array with paragraph data that also specifies the React component  
45   // we'll use to render it.  
46   const mappedParagraphs = paragraphs  
47     // Add a component key that defines the component using the following  
48     // naming convention: ParagraphTypeProvider  
49     .map(paragraph => {  
50       if (paragraph && paragraph.__typename) {  
51         const componentType = paragraph.__typename.replace("paragraph__", "")  
52         paragraph.provider = componentMap[componentType]  
53       }  
54       return paragraph  
55     })  
56   // Filter out paragraph types we don't yet have a mapped component for.  
57   .filter(paragraph => {  
58     return paragraph.provider !== undefined  
59   })  
60   return mappedParagraphs  
61 }
```

Wrapping it all up

```
3
4 import paragraphsToComponents from "../utils/paragraphsToComponents"
5
6 import DefaultLayout from "../components/layouts/DefaultLayout/DefaultLayout"
7
8 // Bundled Drupal component styles that will be used by Gatsby
9 import "../styles/jsonrendered.bundle.css"
10 import { Helmet } from "react-helmet"
11 import metadataToMetatag from "../utils/metadataToMetatag"
12
13 const Page = ({ data, pageContext }) => {
14   const expanded = data.nodeDecoupledPage.field_decoupled_footer_expanded
15   const paragraphs = data.nodeDecoupledPage.relationships.field_decoupled_components
16
17   let paragraphComponents = paragraphs && paragraphsToComponents(paragraphs)
18
19   return (
20     <DefaultLayout expanded={expanded}>
21       <Helmet>
22         {pageContext.metatag?.map(item => {
23           return metadataToMetatag(item.attributes)
24         })}
25       </Helmet>
26       {paragraphComponents &&
27         paragraphComponents.map((paragraph, index) => {
28           return (
29             <div key={`paragraph_${index}`}>
30               {paragraph.provider({
31                 paragraph: paragraph,
32                 index: index,
33               })}
```

Justifying a Decoupled Build

Struggling with cost vs benefit

Front End Structure

A single codebase serving multiple front ends

Hydrating Gatsby

How we got data from Drupal to the frontend

Workflow Changes

The impact of introducing static builds

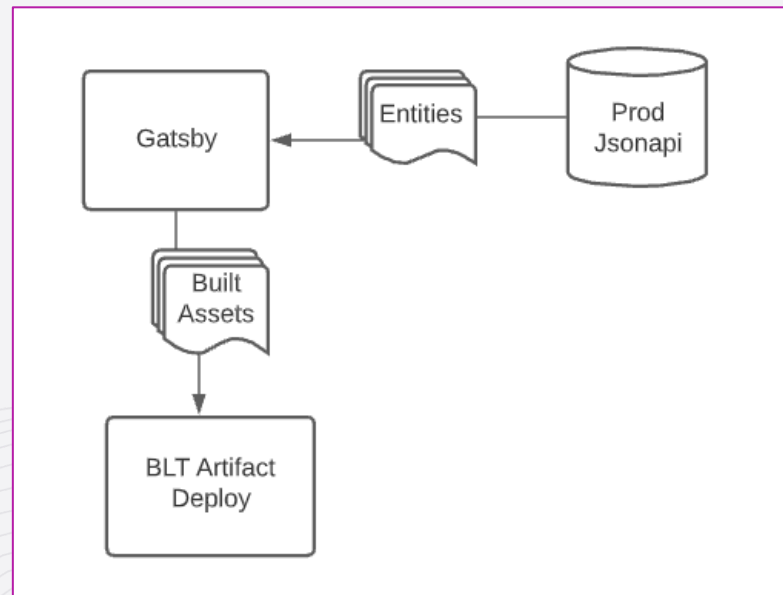
The Blurring Line Between Front Ends

The impact of managing multiple component libraries



Building the site

- `npm ci && npm run build`
- `cp -R gatsby/public docroot/public`
- `composer install && vendor/bin/blt artifact:deploy`



Serving Gatsby and Drupal

```
#####  
# Gatsby Content  
  
# Check for static path.  
RewriteCond %{REQUEST_URI} ^/public/static [NC]  
RewriteCond %{REQUEST_URI} ^/static [NC,OR]  
RewriteRule .* - [E=GATSBY_STATIC:1,E=GATSBY:1,L]  
  
# Check for anything else under public  
RewriteCond %{REQUEST_URI} ^/public [NC]  
RewriteRule .* - [E=GATSBY:1,L]  
  
# Grab the path and store it in %1  
RewriteCond %{REQUEST_URI} ^(.*)$  
# Check if the requested path is a directory under public.  
RewriteCond %{DOCUMENT_ROOT}/public/%1 -d  
# Ensure that index.html exists in that directory.  
RewriteCond %{DOCUMENT_ROOT}/public/%1/index.html -f  
# Serve index.html  
RewriteRule ^(.*)$ public/$1/index.html [E=GATSBY:1,L]
```

Caching Considerations

```
204 # Set cache tags.  
205 Header set X-Drupal-Cache-Tags "gatsby" env=GATSBY  
206  
207 # Apply cache control.  
208 Header set cache-control "public, max-age=900, must-revalidate" env=GATSBY  
209 Header set cache-control "public, max-age=31536000, immutable" env=GATSBY_STATIC  
210  
211 <FilesMatch "(\\.js|\\.css)$">  
212 | Header set cache-control "public, max-age=31536000, immutable" env=GATSBY  
213 </FilesMatch>  
214  
215
```


Other Workflow Changes

Implemented a "magic button" to trigger Gatsby builds

Live previews using Gatsby Cloud

- We decided a button to trigger a deployment was good enough for the needs of the content team.
- Contributed back a Bitbucket plugin to build_hooks module that triggers a CI pipeline
- Live previews enabled in production, not yet a critical part of workflow
- Appreciated the patience shown by our marketing team as this workflow evolved...

Justifying a Decoupled Build

Struggling with cost vs benefit

Front End Structure

A single codebase serving multiple front ends

Hydrating Gatsby

How we got data from Drupal to the frontend

Workflow Changes

The impact of introducing static builds

The Blurring Line Between Front Ends

The impact of managing multiple component libraries

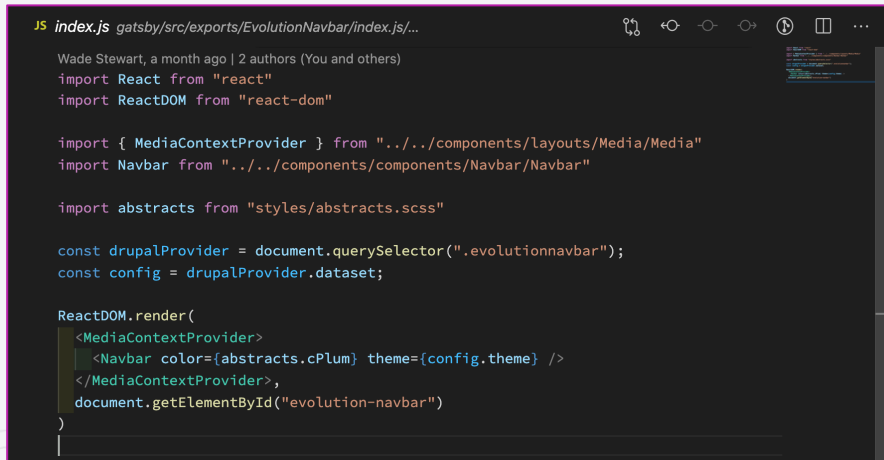


A Drupal Sandwich with React Bread



Syndicating React Header and Footer to Drupal

- Created 'exportable' version of header and footer components
- Used webpack configuration that used header and footer as entry points.
- Bundle components into dist directory in Drupal theme



```
JS index.js gatsby/src/exports/EvolutionNavbar/index.js/...
Wade Stewart, a month ago | 2 authors (You and others)
import React from "react"
import ReactDOM from "react-dom"

import { MediaContextProvider } from "../../components/layouts/Media/Media"
import Navbar from "../../components/components/Navbar/Navbar"

import abstracts from "styles/abstracts.scss"

const drupalProvider = document.querySelector(".evolutionnavbar");
const config = drupalProvider.dataset;

ReactDOM.render(
  <MediaContextProvider>
    <Navbar color={abstracts.cPlum} theme={config.theme} />
  </MediaContextProvider>,
  document.getElementById("evolution-navbar")
)
```

Component Module

Simplifies Drupal integration

- Define configuration in .yaml file alongside js
- Derives component block from config
- Can also specify form configuration
- Form config values added to markup as data attributes

```
! evolutionnavbar.component.yaml docroot/themes/custom/bo

You, seconds ago | 2 authors (Wade Stewart and others)
name: Evolution Navbar
description: 'Evoution Navbar'
type: 'block'
js:
  'dist/navbar.bundle.js' : {}
dependencies:
  - bounteous/react
template: 'evolutionnavbar.html'
form_configuration:
  theme:
    type: select
    title: "Theme"
    options:
      '': 'Dark Theme'
      'theme-light': 'Light Theme'
    default_value: ''

# evolutionnavbar.html is just:
<div id="evolution-navbar"></div>
```

Configure block ☆

[Configure block](#) [Translate block](#) [Devel](#)

[Admin](#) » [Structure](#) » [Block](#)

Block description: Evolution Navbar

Title *

Evolution Navbar Machine name:

This field supports tokens. [Browse available tokens.](#)

☐ Display title

▼ **COMPONENT SETTINGS**

Theme

Dark Theme ▼

Visibility

Content type	Content type
Webform	<input type="checkbox"/> Article
Content types	<input type="checkbox"/> Blog
Not restricted	<input type="checkbox"/> Case Study
Pages	<input type="checkbox"/> Client
Not restricted	<input type="checkbox"/> Client Quote
	<input type="checkbox"/> Client Resource Page
	<input checked="" type="checkbox"/> Client Story
	<input type="checkbox"/> Components Page

Making GraphQL Data Available Outside of Gatsby

- Initially hardcoded menu data
- Now importing cached menu data from Gatsby build
- Feels a bit fragile, but working nicely thus far
- Gatsby build needs to run before Drupal build

JS index.js gatsby/src/exports/EvolutionNavbar/index.js/...

Matthew Ramir, 2 months ago | 4 authors (Matthew Ramir and others)

```
import React from "react"
```

```
import ReactDOM from "react-dom"
```

```
import { MediaContextProvider } from "../../components/layouts/Media/Media"
```

```
import abstracts from "styles/abstracts.scss"
```

```
// Grab cached data from disk.
```

```
import NavigationProvider from "../../components/provider/menu/NavigationProvider"
```

```
import templateData from "../../public/page-data/template/page-data.json"
```

```
// Use 2nd query because... reasons?
```

```
const queryHash = templateData.staticQueryHashes[1]
```

```
const data = require(`../../public/page-data/sq/d/${queryHash}.json`)
```

```
const drupalProvider = document.querySelector(".evolutionnavbar")
```

```
const config = drupalProvider.dataset
```

```
ReactDOM.render(()
```

```
  <MediaContextProvider>
```

```
    <NavigationProvider
```

```
      data={data.data}
```

```
      color={abstracts.cPlum}
```

```
      theme={config.theme}
```

```
    />
```

```
  </MediaContextProvider>,
```

```
  document.getElementById("evolution-navbar")
```

```
)
```

Brian Perry, 5 months ago • BCE-100: initial export of placeholder header and...

Why can't I just use this component?

React Components vs. Drupal Components

- Felt like an arbitrary divide to wider team
- Made ongoing design discussions challenging
- Need to create more React versions of Drupal components to support design

Possible improvements:

- Syndicate Drupal markup to Gatsby
 - We're doing this, but it is a little dirty...
- This is a solid case for using native web components

The screenshot shows a website section with a blue gradient background. At the top, the heading "Our partnerships accelerate innovation" is displayed in white. Below it, a sub-headline reads: "We elevate brand experiences through our technology partnerships and unparalleled platform expertise". A grid of partner logos follows, including Akeneo, Google, Acquia, Adobe, Magento, and Shopify Plus. A button labeled "VIEW ALL PARTNERS" is centered below the logos. A quote from Romain Apert, Global Chief Information Officer at Mars Wrigley, is featured in a white box on the right side of the section. The quote praises Bounteous for its role in launching the TREAT TOWN™ concept. The Bounteous logo is visible in the bottom left corner of the page.

Our partnerships accelerate innovation

We elevate brand experiences through our technology partnerships and unparalleled platform expertise

akeneo Google ACQUIA Adobe Magento shopifyplus

[VIEW ALL PARTNERS](#)

“Bounteous was instrumental in working with Mars to bring the TREAT TOWN™ idea to life and was uniquely capable to get this first-of-its-kind concept and experience launched in time for the holiday season. They brought creative and technology expertise to the table in record time to meet consumer needs, cementing a powerful partnership. TREAT TOWN™ puts the celebration back in Halloween, thanks to Mars and Bounteous.

Romain Apert
GLOBAL CHIEF INFORMATION OFFICER
MARS WRIGLEY

An iterative approach **doesn't eliminate all the challenges** of decoupling.

In fact, it can introduce some new ones.

But it can be an effective way to prove the fit for your organization.

For us, it was the difference between decoupling now or decoupling never.

Contribution Week

Visit:

<https://drupalcontributions.opensocial.site>

Decoupled Menus Mentored Contribution

Today at 2 PM

bounteous

Generic Drupal Web Components

[View](#) [Edit](#) [Version control](#) [View history](#) [Maintainers](#)

By [brianperry](#) on 14 March 2021, updated 11 April 2021



This project is not covered by Drupal's [security advisory policy](#).

Inspired by [@generic-components/components](#) this project aims to create a library of generic web components that are accessible, framework agnostic, possible to style, and easy to use with data provided by Drupal.

A demo of the current components can be found at: <https://gdwc.netlify.app/> and you can also watch a [brief project overview video](#).

Usage

Via NPM

Components can be installed via NPM (View [example on Codesandbox](#)):

```
npm i @gdwc/components
```

And imported in your code via ES imports:

```
import '@gdwc/components/menu';
```

Via CDN

Alternatively you can load the components from a CDN and drop them in your HTML file as a script tag. (View [example on Codepen](#))

Maintainers for Generic Drupal Web Components

[View all committers](#)
[View commits](#)

Issues for Generic Drupal Web Components

To avoid duplicates, please search before submitting a new issue.

[Search](#)

[Advanced search](#)

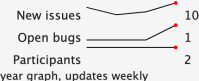
All issues

16 open, 24 total

Bug report

1 open, 1 total

Statistics



Thank you.

Q&A

(Also, we'd love to hear your session feedback)

Brian Perry

Front End Architect

brian.perry@bounteous.com
@bricomedv / brianperry.dev

Matthew Ramir

Sr. Web Developer

Matthew.ramir@bounteous.com
@crasx