



Kickstart Your Drupal Build with Sous and Emulsify

MidCamp 2023

APRIL 27, 2023



Agenda

- 1 Why Use Sous & Emulsify? Practical solutions to Common Problems
- 2 A (brief) introduction to Component Driven Development
- 3 Example component development

Our team



Laura Johnson

Senior Engineer



Callin Mullaney

Senior Engineer



Amanda Luker

*Sr. Front End
Engineer*



Brian Goldstein

Technical Strategist

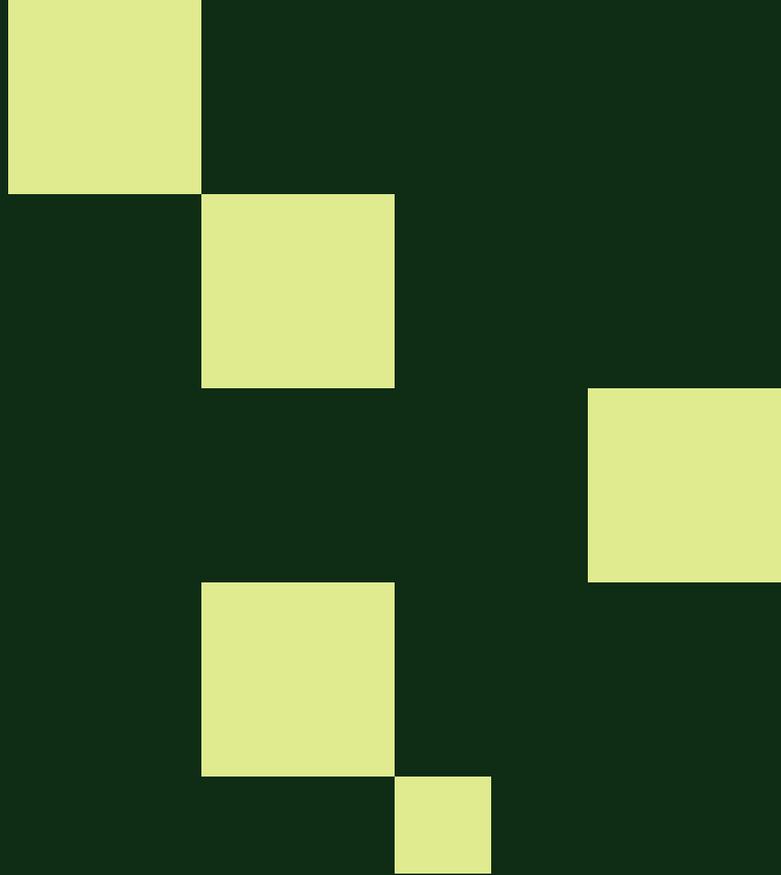
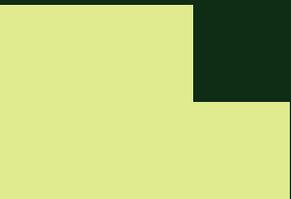
GLOSSARY

We'll use these terms a lot today:

- **Emulsify**: Design system and component library
- **Storybook**: A front-end workshop for building components
- **Sous**: A starter kit for Drupal that comes with an Emulsify-based theme
- **Compound**: A library of pre-built components

PART 1

Practical Solutions



Ninety Nine Problems...



COMMON PROBLEMS

One Size Fits None

Bloated Distributions

It often seems like the only options are to use a completely blank install profile or to select an over-opinionated distribution with a lot of bloat, overhead, or old practices baked in.

Workflow Disconnect

Good luck layering on a workflow that accounts for local, development, and production environments and release management

What's my name again?

The hardest part of most builds can just be getting everyone involved to use the same name when they are talking about or working on the same thing.

Herky-Jerky

Handing off tickets between engineers, or trying to figure out which milestone on a feature comes first leads to a lot of "Two Step Forward, One Step Back" progress.

Sous & Emulsify Solve Many Problems

Some of our favorite things about them

PRACTICAL SOLUTIONS

We thought of it, so you won't need to.

Minimal Assumptions

Sous makes minimal assumptions about what you'll need in a build, but gives enough of a starting point that you're not staring at a complete blank.

Workflow Support

With updated build commands and hosting supports on the way, Sous can help you get all of your environments spun up and synced faster.

Storybook

Integrating Storybook into Emulsify provides a one-stop shop for clients, designers, engineers, and site builders to reference for naming conventions, testing, and what configurations might look like.

Separate Concerns

As a tool for component driven development, Emulsify helps teams work on the front and back ends independently from each other, improving accuracy and efficiency.

PART 2

A (brief) introduction to Component Driven Development

COMPONENT DRIVEN DEVELOPMENT

What is it?

Component Driven Development (CDD) breaks sites and applications into discrete “features” that can be integrated with the CMS, and tested independently from one another.

- **Component:** any part of a website that has a front-end display, a configurable back end, and/or transmits data.
- Common Components are things like Menus, Forms, Buttons, Sliders, or Tabs.

Why do things this way?

- Clarity in User Stories, Acceptance Criteria, and Estimation
- Separation of efforts
- Reduce repetition
- “Lego” complexity
- Isolates testing domains
- Enables lots of flexibility for content editors

Streamline Project Setup

One command to kickoff your project!

```
> composer create-project fourkitchens/sous-drupal-project [PROJECT_NAME] --no-interaction
```

Composer: install all Drupal with all dependencies

Additional scripts:

- **Create** new custom theme based on Emulsify
- **Install** all theme dependencies
- **Build** Storybook
- **Run** the Drupal install script
- **Block** the superuser and create a new admin user
- **Log in** as the admin user

Content Structure Appearance Extend People Reports Configuration

← Edit sous_chef

[Sous Project](#)

sous_chef

- [View](#)
- [Roles](#)
- [Shortcuts](#)
- [Edit](#)
- [Login history](#)

Member for

56 seconds

Tools

- [Add content](#)

Administration

- [All types](#)



Content Structure Appearance Extend People Reports Configuration

← Edit: Welcome to your Drupal project featuring Emulsify!

[Sous](#)
FEATURING *Emulsify*

Example Components

[View](#) [Edit](#) [Delete](#) [Revisions](#)

Welcome to your Drupal project featuring Emulsify!

The Sous Demo has been installed and is available for you to browse, modify, or fork for your own project. It includes:

- **Sous Builder** contrib module with basic configuration for a homepage and basic page content, simple paragraphs architecture, and some block layout config that sets up our general page template.
- **Sous Demo** custom theme generated from EmulsifyCLI. It contains pre-wired templates as well as components installed from the [Compound component library](#).

If you'd like to modify the sousedemo custom theme run the following commands to get it's storybook instance up and running

- `cd web/themes/contrib/sousedemo`
- `nvm use`
- `npm run develop`

```
> cd [PROJECT_NAME] && lando sous-demo-install
```

Lando command: run script that installs default content module & theme

PART 3

Working through an Example Component

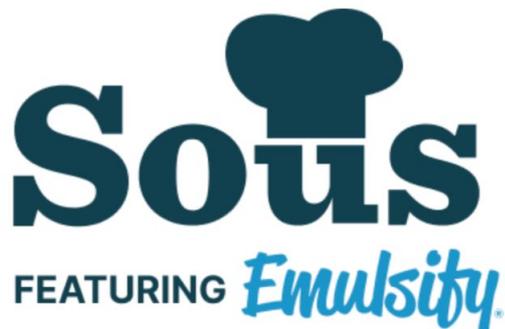
WALK THROUGH

What Are We Building today?

For today's demo, we'll be stepping through a component that comes included, the *Text with Media* component.

We'll also go over some good guidance when you go to build your own components!

Text with Image component



Morbi condimentum, sapien id consectetur mattis, nisl ligula consectetur augue, in euismod sapien purus quis magna. Pellentesque dictum sapien vitae consequat fringilla. Ut ullamcorper congue mi, varius vestibulum odio elementum nec. In a vehicula massa, in rhoncus neque. Nunc sagittis magna sit amet leo dignissim egestas. Mauris id augue cursus, porta justo sit amet, scelerisque nisi.

Cras in ipsum vel ante dapibus vehicula nec id ipsum. Interdum et malesuada fames ac ante ipsum primis in faucibus. Aenean mattis elit nec metus mattis, id vulputate mi rhoncus. Nullam sodales ac nisi quis sollicitudin.

WALK THROUGH

Launch Storybook

```
> lando compound-install
```

```
> npm run theme-watch
```

```
Storybook 6.5.15 for HTML started
```

```
9.18 s for manager and 9.61 s for preview
```

```
Local: http://localhost:6006/
```

```
On your network: http://192.168.2.63:6006/
```

```
A new version (7.0.6) is available!
```

```
Upgrade now: npx storybook@latest upgrade
```

```
Read full changelog: https://github.com/storybookjs/storybook/blob/next/CHANGELOG.md
```



Find components

- BASE
 - Colors
 - Palettes
 - Motion
 - Typography
- ATOMS
 - Button
 - Forms
 - Images
 - Links
 - Lists
 - Tables
 - Text
 - Videos
 - Wide
 - Full
- MOLECULES
 - Cards
 - Carousel
 - Contact Information
 - CTA
 - Event List
 - Event Teaser
 - Logo
 - Menus
 - Sidebar Blocks
 - Sidebar Navigation
 - Status
 - Tabs
 - Teaser

Canvas Docs
🔍 🔍 🔍
🏠 📄 📑 🗑️
🔗 📄 📄

Voting for democracy and a better life

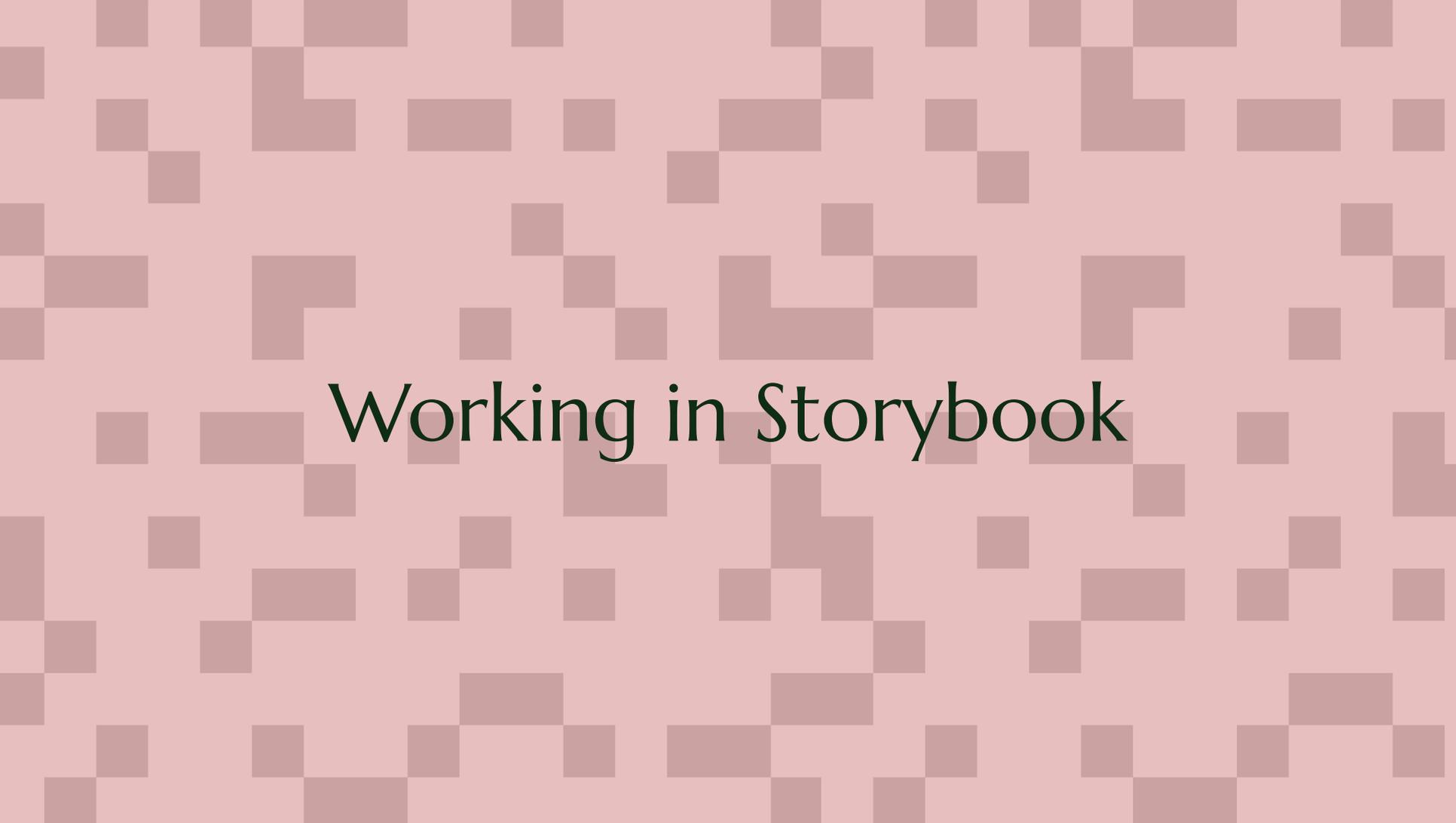


In the leadup to the midterm elections, pundits predicted a red wave, even a tsunami, based on polls, historical precedent, and steep gas and grocery prices. But I had my doubts. I spent the weeks before the elections talking to voters and traveling on the AFT Votes bus, rolling through a dozen states with more than 50 stops. In a year when kitchen table issues, democracy and our freedoms were on the ballot, many people told me that the elections came down to a choice between, on the one side, election deniers and extremists stoking fear, and on the other, problem-solvers working to help the country move forward. Many races were close, but Americans turned the tide from a red wave to a swell of support for progress and problem-solvers.

Accessibility
Controls (2)
Actions
🔗 🗑️

Name	Control
Layout	<div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> Choose option... <ul style="list-style-type: none"> <li style="background-color: #007bff; color: white; padding: 2px 5px;">✓ title_first <li style="padding: 2px 5px;">image_first </div> <div style="display: flex; gap: 10px;"> False True </div>
Inverted	

The Storybook UI



Working in Storybook

WALK THROUGH

Working in Storybook

Component Files

- a. .twig
- b. .yml
- c. .scss and/or .js
- d. .stories.js



WALK THROUGH

.twig file

```
{% set text_with_image_base_class = 'text-with-image' %}

{% set text_with_image_attributes = {
  'data-image-position': text_with_image_position|default('left'),
  'class': bem(text_with_image_base_class, text_with_image_modifiers,
text_with_image_blockname)
} %}

<div {{ add_attributes(text_with_image_attributes) }}>
  {# Content #}
  <div {{ bem('content', [], text_with_image_base_class) }}>
    {# Heading #}
    {% if text_with_image_heading %}
      {% include "@atoms/text/headings/heading.twig" with {
        heading: text_with_image_heading,
        heading_level: '2',
        heading_blockname: text_with_image_base_class,
      } %}
    {% endif %}
```

WALK THROUGH

.twig file

```
{% include "@atoms/text/text/text.twig" with {
    text_content: text_with_image_text,
    text_blockname: text_with_image_base_class,
} %}

{# optional link #}

{% if text_with_image_link_url %}

    {% include "@atoms/link/link.twig" with {
        link_content: text_with_image_link_content,
        link_url: text_with_image_link_url,
        link_style: 'arrow-right',
    } %}

{% endif %}

</div>

<div {{ bem('image', text_with_image_modifiers,
text_with_image_base_class) }}>

    {% block text_with_image_image %}

        {% include "@atoms/images/image/image.twig" %}

    {% endblock %}

</div>

</div>
```



WALK THROUGH

.yml file

```
text_with_image__underline: 'Overline / Eyebrow'  
text_with_image__heading: 'Master's in Biological  
Sciences'  
text_with_image__text: '<p>The Department of Biological  
Sciences maintains an active graduate program leading to  
the master of science degree in biological sciences,  
emphasizing independent research supported by advanced  
course work.</p>'  
text_with_image__link__content: 'Example link here'  
text_with_image__link__url: '#'
```



WALK THROUGH

.stories.js file

```
// Twig templates
import textWithImageTwig from './text-with-image.twig';

// Data files
import imageData from
'../../01-atoms/images/image/image.yml';
import textWithImageData from './text-with-image.yml';

/**
 * Storybook Definition.
 */
export default { title: 'Molecules/Text With Image' };

export const textWithImageTwig = () =>
TextWithImage(textWithMediaData);
```

Sous

Find components

- BASE
 - Breakpoints
 - Colors
 - Motion
 - Spacing
 - Typography
- ATOMS
 - Button
 - Forms Inputs
 - Images
 - Links
 - Lists
 - Tables
 - Text
 - Videos
- MOLECULES
 - Logo
 - Page Title
 - Status
 - Tabs
 - Text
 - Text With Image**
 - Video
- ORGANISMS
 - Navigation
 - Site
- TEMPLATES
 - Container
 - Grid
 - Place Holder

Canvas Docs



Master's in Biological Sciences

The Department of Biological Sciences maintains an active graduate program leading to the master of science degree in biological sciences, emphasizing independent research supported by advanced course work.

[Example link here](#)

Accessibility (1) Controls (5) Actions

1 Violations 9 Passes 0 Incomplete

Highlight results

> Elements must meet minimum color contrast ratio thresholds

Tests completed

What our component looks like in Emulsify right now

Wiring our component

Content Body *

Text with Image Collapse

Text *

Paragraph **B** *I* Source

Text format Basic HTML [About text formats](#)

Media *

No media items are selected.

Add media

Media Alignment *

Left

Before Wiring: Drupal Text with Media paragraph form

Content Structure Appearance Extend People Reports Configuration

Back to site Administration / Structure / Paragraphs types / Text with media Shortcuts sous_chef

Manage display ☆ + Add field group

Edit Manage fields Manage form display **Manage display**

Field	Label	Format	Show row weights
✚ Text	- Hidden -	Default	
✚ Media	- Hidden -	Rendered entity	Rendered as Paragraph Text with Image ⚙️
✚ Media Alignment	- Hidden -	Key	

Disabled

No field is hidden.

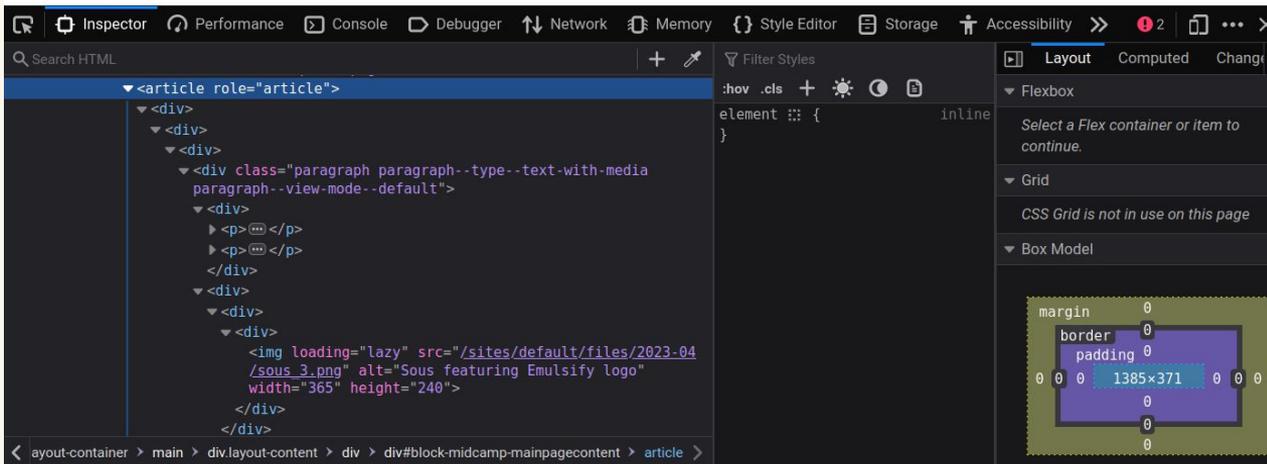
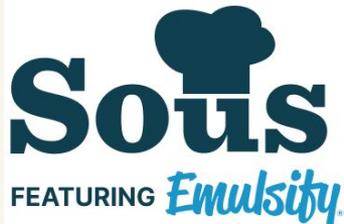
▼ Custom display settings

Save

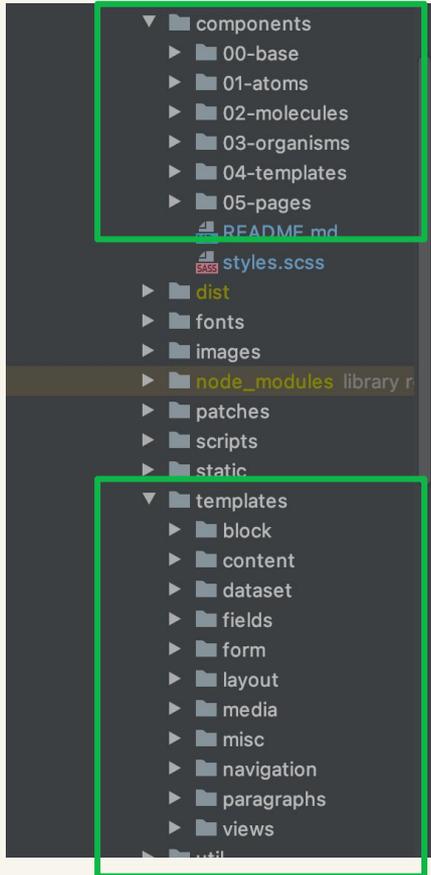
Verify the field output in the Manage Display is set to what our component expects to render.

Morbi condimentum, sapien id consectetur mattis, nisl ligula consectetur augue, in euismod sapien purus quis magna. Pellentesque dictum sapien vitae consequat fringilla. Ut ullamcorper congue mi, varius vestibulum odio elementum nec. In a vehicula massa, in rhoncus neque. Nunc sagittis magna sit amet leo dignissim egestas. Mauris id augue cursus, porta justo sit amet, scelerisque nisi.

Cras in ipsum vel ante dapibus vehicula nec id ipsum. Interdum et malesuada fames ac ante ipsum primis in faucibus. Aenean mattis elit nec metus mattis, id vulputate mi rhoncus. Nullam sodales ac nisi quis sollicitudin.



Before Wiring: What our component looks like in Drupal

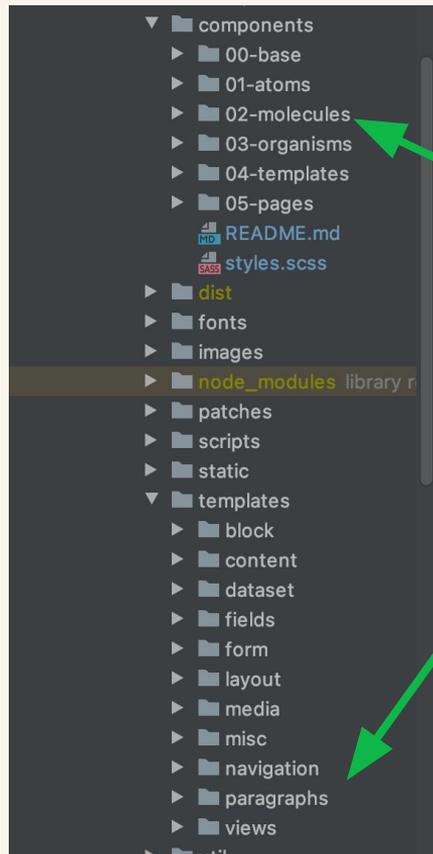


File Structure

Emulsify templates

Drupal templates

File Structure



Emulsify twig template: **text-with-image.twig**

Drupal twig template: **paragraph--text-with-media.html.twig**

```
{% embed "@molecules/text-with-image/text-with-image.twig" with {
  text_with_image_text: content.field_text,
  text_with_image_position: content.field_media_alignment.0['#markup'],
}%}

{% block text_with_image_image %}
  {{ content.field_media }}
{% endblock %}

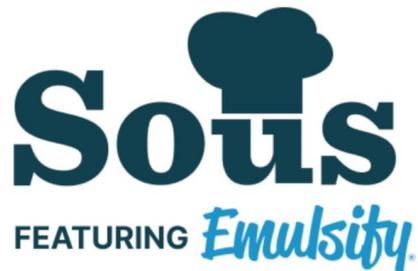
{% endembed %}
```

The screenshot displays the EmulsiFY editor interface. On the left, a component tree lists various elements like Tables, Text, Videos, MOLECULES, Logo, Page Title, Status, Tabs, Text, Video, ORGANISMS, Navigation, Site, TEMPLATES, Container, Grid, and Place Holder. The 'Text With Image' component is selected.

The main preview area shows a component with a yellow header bar, a large image of a modern building interior, and a text block titled "Master's in Biological Sciences". Below the text is a link "Example link here" with a right-pointing arrow.

The bottom half of the image shows the browser's developer tools. The HTML pane shows the component's structure, including a root div with a class "container" and a nested "text-with-image" div. The CSS pane shows the styles for the selected element, including "display: block;" and "width: 100%;". The Box Model pane shows a visual representation of the element's dimensions, with a width of 1149px and a height of 432px.

Before Wiring: What our component looks like in EmulsiFY



Morbi condimentum, sapien id consectetur mattis, nisl ligula consectetur augue, in euismod sapien purus quis magna. Pellentesque dictum sapien vitae consequat fringilla. Ut ullamcorper congue mi, varius vestibulum odio elementum nec. In a vehicula massa, in rhoncus neque. Nunc sagittis magna sit amet leo dignissim egestas. Mauris id augue cursus, porta justo sit amet, scelerisque nisi.

Cras in ipsum vel ante dapibus vehicula nec id ipsum. Interdum et malesuada fames ac ante ipsum primis in faucibus. Aenean mattis elit nec metus mattis, id vulputate mi rhoncus. Nullam sodales ac nisi quis sollicitudin.

```

<!-- BEGIN OUTPUT from
'themes/contrib/sousdemo/templates/paragraphs/paragraph--
text-with-media.html.twig' -->
<div data-component-theme="white" data-component-width="stan
ard" class="container container--with-bg" data-component-
alignment="center">
  <div class="container__inner">
    <div data-image-position="left" data-component-focus="eq
al" class="text-with-image"> { flex } == $0
      <div class="text-with-image__content"> </div>
      <div class="text-with-image__image"> </div>
    </div>
  </div>
<!-- END OUTPUT from
'themes/contrib/sousdemo/templates/paragraphs/paragraph--
text-with-media.html.twig' -->
  
```

```

Styles
Computed Layout Event Listeners DOM Breakpoints Properties Accessibility >>
Filter :hov .cls + [ [ ]
element.style {
}
@media (min-width: 47.9375rem)
.text-with-image[data-image-position=left][data-component-focus=equal] {
  flex-direction: row-reverse;
}
.text-with-image[data-image-position=left][data-component-focus=equal] {
  flex-direction: column-reverse;
}
[data-component-alignment=center][class*=__inner]>* {
  margin-inline: > auto;
}
[data-component-width=standard][class*=__inner]>* {
  width: 100%;
  max-width: var(--content-width);
}
  
```

After Wiring

WIRING GROUND RULES

Going from Simple to Complex

Because Drupal websites (and Emulsify) often have components nesting or looping within other components, it can quickly get more complicated.

Get familiar with these Twig terms:

- Include
- Embed
- Extend
- Blocks

WIRING GROUND RULES

Looping in template files

```
{% embed "@molecules/accordion/accordion.twig" with {  
  accordion__heading : content.field_heading.0,  
}%}  
  
{% block accordion__items %}  
  {{ content.field_accordion_items }}  
{% endblock %}  
  
{% endembed %}
```

Loops are a necessary part of templating that allow us to create lists or grid-type layouts.

Whenever a template requires a loop, the loop should iterate over a separate component or a twig partial.

Never attempt to preprocess values of a loop and send through a `include/embed`!

WIRING GROUND RULES

Looping in template files

```
{% embed "@molecules/accordion/accordion.twig" with {  
  accordion__heading : content.field_heading.0,  
}%}  
{% block accordion__items %}  
  {{ content.field_accordion_items }}  
{% endblock %}  
{% endembed %}
```

paragraph--accordion-item.html.twig

```
{% include "@molecules/accordion/_accordion-item.twig" with {  
  accordion__item__heading: content.field_heading,  
  accordion__item__content: content.field_text,  
}%}
```

WIRING GROUND RULES

Let each system do what it does best

👍 Example:

```
{% embed "@molecules/teaser/  
teaser.twig" %}  
  
{% block teaser_image %}  
  {{ content.field_teaser_media.0 }}  
{% endblock %}  
  
{% endembed %}
```

Using Drupal's markup output can be more cost-effective than rendering all markup through Emulsify. Typical areas where this applies include:

- Contextual Menus
- Webforms
- Views Exposed Forms

However, it's still important to try to implement BEM type classes for consistent styling.

WIRING GROUND RULES

Use Drupal's field displays rather than working with raw data

👍 Example:

```
{% include "@molecules/teaser/teaser.twig" with {
  teaser__body: content.field_teaser,
} %}
```

*Not: `teaser__body:node.field_teaser.0.value`

Your first step should always be to configure the fields in the user interface. Because it is:

- **Faster**
- Makes Drupal Twig **more clear**.
- Default **#cache** feature.
- **Applies** default field values.
- No tricks needed for **Multilingual** values.

WIRING GROUND RULES

Good guidance

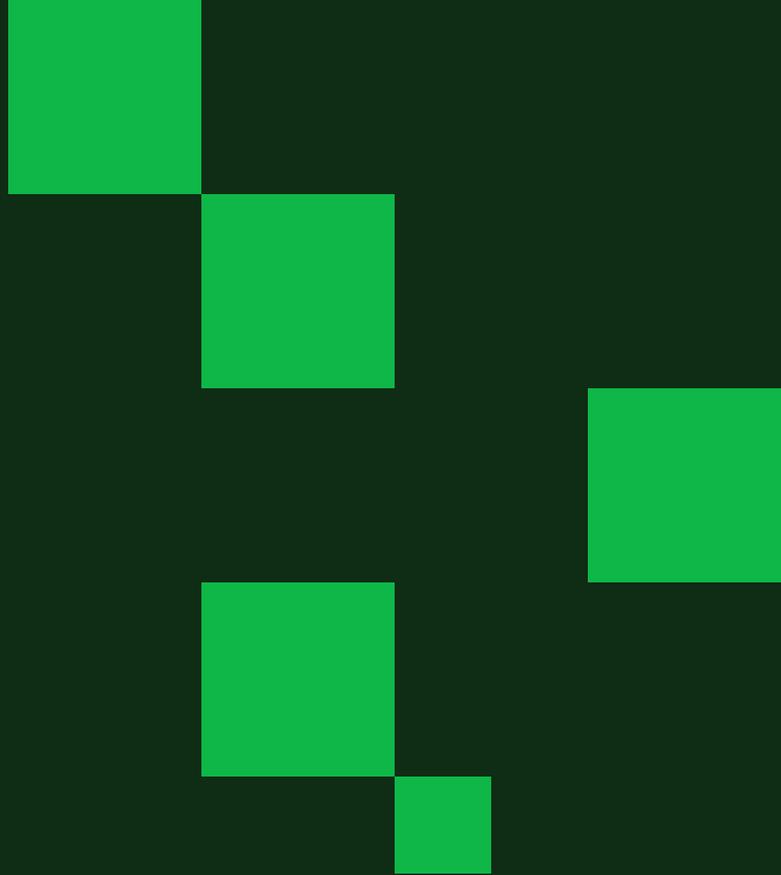
- **Render Drupal's arrays**-of-doom before sending to Emulsify
- **Use** Drupal's responsive image system
- **Let** Drupal render Forms
- Only override Views templates for layout reasons
- **Use view-modes** where possible

Thank you!



APPENDIX 1

Instructions





Commands we used today:

a single command to get a working Drupal site:

```
composer create-project  
fourkitchens/sous-drupal-project [site-name-here]  
--no-interaction
```

Install default content and demo theme:

```
lando sous-demo-install
```

Install compound components:

```
lando compound-install
```

Install Storybook:

```
npm run theme-watch
```

APPENDIX 2

Further Resources and Documentation



Find out more about

- Sous
 - [Sous repository](#)
 - [Default content module repository](#)
 - [Drupal Project](#)
- Emulsify
 - [Site](#)
 - [Drupal Project](#)
 - [Documentation](#)
- [Storybook](#)
- [Compound](#)
- [Four Kitchens](#)
- [Twig variable reference](#) (from Sarah Carney)
- [Twig Tweak](#) / [Twig Tweak cheat sheet](#)