Expanding On Site Building With Components To Truly Achieve No-code Drupal

Peta Hoyes
Tag1 Consulting
Chief Operating Officer
drupal.org/u/phoyes

Fabian Franz
Tag1 Consulting
VP Software Engineering
drupal.org/u/fabianx
Tag1 Consulting
The Team Global Leaders Trust to Ensure Their Success

Peta Hoyes @phoyes
Partner/COO
- Leads Operations, Delivery, Finance
- Mechanical Engineer With a Passion for Great Product Design and Art
- Created One of the First Hyper-Local Communities, New York Online In 1993

Fabian Franz @fabianfranz
VP, Software Engineering
- 1 of 3 Drupal (D7) Framework Managers
- Performance & Scalability Expert
- Architected Code Used by 2% of The Internet & 10% of Top 100k Sites
## About Tag1

<table>
<thead>
<tr>
<th>The New York Times</th>
<th>The Linux Foundation</th>
<th>ALJAZEERA</th>
<th>Michigan</th>
<th>The White House</th>
</tr>
</thead>
<tbody>
<tr>
<td>FORTIVE</td>
<td>tern</td>
<td>Symantec</td>
<td>Stanford University</td>
<td>Confidential Forum 500</td>
</tr>
<tr>
<td>Capgemini</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FOREO</td>
<td>ACLU</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
#Tag1TeamTalks and Blogs

Dive Deeper Into Hot Topics in Drupal

- **Enabling Declarative Components Everywhere: Drupal + Web Components Out of the Box**
  - #Tag1TeamTalk
  - Tag1.com/Web-Components

- **A Deep Dive into End-to-End Encryption in Yjs - Part 1**
  - #Tag1TeamTalk
  - Tag1.com/Yjs

- **Introducing Goose - a highly scalable load testing framework written in Rust**
  - #Tag1TeamTalk
  - Tag1.com/Goose
What We’ll Cover

➔ Why Low-Code, No-Code Technology
➔ Isn't Drupal There Already?
➔ Drupal's Low/No-Code Strengths and Weaknesses

II. Closing the Gap Between Perception & Reality of No-Code Drupal:

➔ The Tailwind comes
➔ Drupal's Theme System: Challenges + Solutions
➔ DEMO!
Why Low-Code, No-Code Technology
Using Visual Interfaces, Simple Logic, and Drag/Drop Instead of Extensive Coding

➔ Build Without Expert Engineering Skills
➔ Enable Business Users To Do Their Jobs Easier/Better
➔ Provides OOTB Functionality
➔ Ensure Cross-Platform Accessibility
➔ Application Lifecycle Management
Isn't Drupal Low/No-Code Already?
Yes, and It Keeps Improving

Site Building
- Content Types & Fields
- Custom lists with Views
- Access Control with Roles & Permissions
- site navigation with menus

No-Code Drupal

Custom Layouts, Pages and Styles
- Layout Builder for Templated Content and Case-Based Overrides
- One-off Landing Pages Not Tied to a Content Type or Structured Content
- HTML / CSS in WYSIWYG Editors

Low-Code Drupal
Drupal's High, Low, No-Code Kung Fu

Awesome Power Requires Jedi Training (and Responsibility)

+ Extremely Powerful, Secure, & Customizable
+ The Ultimate Cross-functional Collaboration Tool

But...

- Comparatively Complex
- Sizeable Upfront Learning Curve (Theming!)
Self-Help Ends at The Theme Layer

Drupal's Theming Is Complicated

➔ Drupal's Base Themes Look Outdated

➔ Commercial Themes for Drupal Are Pretty Bad

➔ Custom Themes Are Expensive and Need Expertise
Closing the Gap Between Perception & Reality of No-Code Drupal
The Tailwind Comes
We have 2021! A Whole New World!

Tailwind, Alpine, React, ...

➔ TailwindCSS Is a Game-Changer in the Industry

➔ AlpineJS Allows Interactivity Simpler Than Jquery

➔ Modern Browsers Make Styling Easier Than Ever
Tail-What?

What is TailwindCSS?

➔ TailwindCSS is a "Styling Language" That Bridges the Gap Between Developer and Designer

➔ Everyone Can Learn It

➔ Backend Developers Can Finally Style Things

➔ Designers Can Finally Constraint Their Layouts
How Do You Use It?

Drupal.org/Project/TailwindCSS

1. Install The Theme

2. Tweak tailwind.config.js With Your "Design System" (Fonts, Colors, Spacing, ...)

3. Style Your Site
How Do I Style My Site?

Example: CelebrateDrupal.org

➔ Completely Open Source Practical Tailwind Theme


➔ Worth Checking Out To See Tailwind In Action
How Do I Style My Site?
Example: Rebuilding-Bartik.OliverDavies.uk
How Do I Style My Site?

Example: Rebuilding-Bartik.OliverDavies.uk

```html
<nav class="flex flex-wrap pb-2 -mt-1 -mx-3 sm:-mx-0 md:p-0">
  <div class="w-full inline-block px-1 mt-1 md:mb-0 sm:w-1/3 sm:pl-0 md:w-auto">
    <a class="block px-3 py-2 rounded-lg text-sm text-black no-underline sm:text-center md:rounded-none md:rounded-t-lg bg-white">
      Home
    </a>
  </div>
</nav>
```
Confused?

 TailwindCSS.com

 ➔ In the Long Term This Is Much Simpler Than Learning All the CSS "Tricks"

 ➔ CSS + Markup Really Go Together, This Removes the Artificial Separation

 ➔ Gain All the Best-Practices Automatically
Overwhelmed?

TailwindCSS.com

➔ It's a New Language, Give It Time. You Learned Site-Building, You Can Learn This!

➔ Live Preview (Wysiwyg) Really Helps, Can Tweak It Till It Works

➔ 1000s of Snippets on the Internet + Commercial Offering As a Great Learning Resource
Underwhelmed? Prefer CSS?

TailwindCSS.com

➔ It's a New Language, Give It Time and Try It Out

➔ Live Preview (WYSIWYG) Really Helps, Can Tweak It Till It Works

➔ Localized Changes Make Maintenance Much Easier

(No Team I Know Was Disciplined Enough to Do Pure BEM)
Too Simple For An Expert?
Experts Build a Site, The Client Maintains It

➔ Code Review Is Simple [Even Possible Finally]

➔ Constraints on What Is Used: Every Design Change Is Easily Visible (as It Changes Tailwind.config.js)
  ➔ You Don't End up With 10 Different Font Sizes, 30 Colors, ...

➔ Components Are Great Building Blocks!
Conclusion

Never Write CSS Again

➔ Can Style a Complete Site Without Writing a Single Line of CSS

➔ Combining Utility Classes => Most Complex Layout

➔ Tailwind Really Wants You to Create HTML With Classes
Drupal's Theme System
Challenges + Solutions
Decoupled Is a Great Fit!

TailwindCSS + React = <3

➔ Decoupled Has Full Control About Markup => Not Low-Code

➔ React Is Nicely Split In Components

➔ But: React Is Not No-Code => Not for Us, Not Here Today.

(However: Really Simple React Without Tooling, Try: Preact + Htm -- Closest to Low-Code React You Can Get)
Drupal's Theme System Challenges

Drupal Is Backend-First :(

➔ Morten DK Since 2011 (And Longer):

I Want To Control All The Markup!!! [And Want No <divs> Everywhere]

➔ Drupal: **Nope :D**
Drupal's Theme System Challenges

Render-Tree / Theme System Is Contrib Friendly

➔ Very Hard to Control the Markup in a Logical Way

➔ Theme Suggestions -> Markup Is Split Across Many Templates All Over the Place

➔ Site Builder (Yes You!) Can Add New Things Arbitrarily
Drupal's Theme System Challenges
Render-Tree / Theme System Is Contrib Friendly

→ Decoupled: Frontend First

→ Drupal: Backend First

=> Let's Make Drupal Frontend First!
Drupal's Theme System Challenges

Let's Make Drupal Frontend First

➔ Components Everywhere - Add Data() Method to All Objects and Create Data That Is Passed to the Theme System

[Amsterdam, 2019] => Has Not Happened, Yet

➔ UI Patterns - https://drupal.org/project/ui_patterns

(Very Patternlab Heavy, Not so Drupally, More for Larger Teams)

➔ Need to Create Yml Files for Every "Theme Hook"
Drupal's Theme System Challenges
Small Steps!

➔ First Attempt (TernBicycles.com - Session From Monday):

➔ Add Tailwind + Alpine

➔ Add Classes Directly to Views, Field Group, Field Formatters, Etc.
Integration With LayoutBuilder

Three ways!

Site Builders Can Now Control Styles Including for Mobile Devices Through Configuration, Not Code

This Module Will Be Open Sourced Soon!
Integration With LayoutBuilder

layout_builder_classes

➔ Amazing Flexibility

➔ Decouple Inner Content From Outer Presentation [Suitable for Translation]

➔ Drawback: Lot's Of Repetitions
Integration With LayoutBuilder

layout_builder_styles

➤ Maps Styles to Classes

➤ Decouple Inner Content From Outer Presentation [Suitable for Translation]

➤ Drawback: All Styles Are Global (Lot's of Checkboxes!)
Integration With LayoutBuilder

layout_builder_components

➔ Wraps Blocks With Components, Free Text Field

➔ Decouple Inner Content From Outer Presentation [Suitable for Translation]

➔ Drawback: Not No-Code [See Next Slides]
Drupal's Theme System Challenges

Small Steps!

➔ Dream: Create a Complex Page Completely via Components

➔ Just Create a Full Page Completely Outside of Drupal

➔ Pure PHP, Platform Agnostic (Decoupled!)
Small Steps

Dream: Create a Complex Site Completely via Components

→ React like:

```javascript
function component__TernTagline($props) {
    $title = $props['children'];
    return <<<EOF
        <span class="text-center font-thin text-gray-700">
            {$title}
        </span>
    EOF;
}
```
Small Steps

Dream: Create a Complex Site Completely via Components

➔ React like:

```javascript
function component__TernAccessoryPage($props) {
    $tagline = component('TernTagline', ['children' => 'Tagline']);
    return <<<EOF
        <div class="flex">
            {$tagline}
        </span>
    EOF;
}
```
Automatic Reloading

Dream: Create a Complex Site Completely via Components

➔ Browser-Proxy Is Setup to Automatically Reload the Page When A Component Changes

➔ As It's Pure PHP, The Performance Is Great

➔ Custom Menu Callback
Small Steps

Dream: Create a Complex Site Completely via Components

➔ It Worked - but Is Not Really No-Code

➔ Integration Then With Layoutbuilder Content Blocks + Sections
  (New Module!)

➔ https://github.com/LionsAd/layout_builder_components - DEV
  Release, Not For Production (out tomorrow)
Integration With LayoutBuilder

Summary

➔ Component => Free Text Field To Call Any component_function

➔ https://drupal.org/project/layout_builder_styles => Define a Style Mapping to Some Classes (Global)

➔ https://github.com/LionsAd/layout_builder_classes => Arbitrarily Define Classes on the Blocks  (out tomorrow)

[Could Be Merged Into layout_builder_styles]
Drupal's Theme System Challenges

Another Dream!

➔ Dream 2:

- You See a Menu
- You Enable the Contextual Link
- You Click "Edit Component"
- You Change Some Classes and Preview Your Changes
- You Save + It's Done! => Wow, Change Implemented.
THE Site-Building Approach to Theming

The Dream

➔ All Changes Are Stored As Configuration and Then Override the Theme Registry Templates

➔ You Can Also Override Theme Suggestions

➔ You Can Easily Map More Complex Entity Fields to the "Component Fields"
THE Site-Building Approach to Theming

So Close: GitHub.com/LionsAd/Component_Library

→ component_library (Out Now!) Allows Programmatic Creation of Twig via UI Including Live Preview and Syntax Highlighting (https://www.drupal.org/project/component_library)

Thx Ivan (@Chi) and Lucas (@heddn) and @tag1consulting!

→ It Is Integrated With ui_patterns for Easy Mapping

→ It's Still Wip and Likely Some Things Will Change
THE Site-Building Approach to Theming

So Close!

➔ We Don't Yet Have the Contextual Links => Magic Naming

➔ Many Templates Are Still Scattered and Would Profit From Being Routed to Larger Components With Much Data()

➔ It's a Great Start Though!
THE Site-Building Approach to Theming

So Close!

➔ DEMO TIME:

➔ Style Things With Tailwind

➔ Make Them Interactive With Alpine
QUESTIONS?
Tag1 Consulting: Follow Us!

Thank You!

Peta Hoyes @phoyes
Partner/COO
➔ Tag1Consulting.com/Blog
➔ YouTube.com/Tag1Consulting

Fabian Franz @fabianfranz
VP, Software Engineering
➔ Twitter.com/FabianFranz