

Hybrid Drupal: Custom Software Development



John Brooks
VP of Engineering
at Sandstorm

Drupal.org: @keepersmith
<https://sandstormdesign.com>

Fun facts about Sandstorm®



Accessibility-certified,
creative technology
agency for 25+ years
for **purpose-driven
brands**

Deep expertise with
**Associations, Global
Non-Profits, Higher Ed &
Healthcare**



Certifications:

CPACC (accessibility), AWS
Cloud, Pantheon, Acquia,
Drupal, DEIB, and WBENC
(women-owned)

15+ yrs working
in Drupal



Interdisciplinary team:
UX, UI, AI, strategy,
technology, analytics

UX practice started by
Ph.D in human factors
in 2006

Sandstorm® Championing Drupal



DrupalCon 2025

Session 1:

[Analytics and Custom
GA4 Reporting](#)

Session 2:

[Bridging Design and
Development with
Storybook & Drupal](#)



DrupalCon Global

10 Design Principles for

Better UX and Accessibility:
<https://youtu.be/f7GoQ5tLwSU>



Drupal MidCamp 2025

Session 1:

Back-End, Front-End
[Unlocking the Hybrid Drupal
Solution for Custom
Software Development](#)

Session 2:

[Leveraging the Best of
Drupal to Support the
World's Greatest Minds](#)



DrupalCon 2024

[Milken Institute +
Sandstorm – Hitting
'Save As' & Transforming
UX, Taxonomy & Search
for a Think Tank Website](#)

Good Work for Good People



THE UNIVERSITY OF
CHICAGO



GOAL OF THIS SESSION

Hands-on Workshop:
Using Drupal and hybrid
javascript framework
components for custom
software development



OUR PARTNERSHIP WITH ALEÄQUO

Aleäquo is a first-of-it's-kind app to access, compare, and manage insurance quotes for thousands of properties and associations.

A digital portal platform for property management companies and association managers, Aleäquo automates the manual processes of gathering property details for insurance quotes. The portal experience includes a rich sign-up process, which securely captures and stores complex data from property managers, while allowing the Aleäquo team to manage data on thousands of properties and associations, which is also integrated to Aleäquo's CRM solution.

The solution also allows users to submit data to obtain insurance documents via an efficient, multi-step process (CertSender).

Our Scope of Work:

- ✓ Design-Thinking Workshops
- ✓ User Research
- ✓ Competitive Analysis
- ✓ Key Messaging
- ✓ User Flows
- ✓ UX Design
- ✓ Usability Testing
- ✓ App Development
- ✓ Web Development
- ✓ Complex Integrations
- ✓ Accessibility Compliance
- ✓ Analytics & Reporting



Custom Software – Why Drupal?

- Developer team and/or client familiarity
- Open-source, out-of-the-box database, user management and control panel. These are features that most custom software needs, and Drupal provides this by default.

Plus:

- ✓ Security / Security Updates
- ✓ Extendability: Building APIs is easier than ever
- ✓ Rich ecosystem of contributed modules
- ✓ New developer ramp-up time is condensed

Why use a JavaScript framework as well?

- Drupal is optimized to allow non-developers to build websites.
- If we want our website to “feel” like an app, there is no avoiding the need for javascript.
- Complex javascript is easier to code with a framework like VueJS or ReactJS.
- Many folks will have different opinions on this topic – that is OK.
- Typescript can be a big advantage, especially for larger development teams.
 - Naturally fewer bugs - good coding practices are enforced by default
 - Naturally auto-documenting (IDEs and Github Copilot enhance efficiency)
 - New developers can be onboarded much faster

Drupal plus a JavaScript framework seems cool. Which framework is best?

01/

VueJS pluses

A bit easier for a “vanilla” javascript developer to understand and learn, as syntax is not so different from standard HTML/Javascript/CSS.

Only a relatively small number of ways to accomplish a goal, leading to less decision paralysis, and fewer coding dead ends / potholes.

02/

ReactJS pluses

Much larger developer community

Many ways to accomplish a goal:

- An incredibly large assortment of open source packages to use and learn from
- Vast numbers of “component frameworks”, state management techniques (built-in context vs redux, etc.)

Knowledge and skills easily translate to React Native

- React Native is very popular for building native mobile apps, and there is heavy crossover with toolsets, coding practices, and packages

03/

Angular pluses

Very opinionated – there is generally only one “right” way to do things.

▼ questionnaire

> modules

> questionnaire-vue

▼ src

▼ Controller

🔗 QuestionnaireDisplayController.php

🔗 QuestionnaireTempProfileController.php

> Form

> Normalizer

> Plugin/rest/resource

> Schema

> Service

▼ templates

📄 questionnaire-display.html.twig

📄 temp-profile.html.twig

{ } package-lock.json

! questionnaire.info.yml

! questionnaire.libraries.yml

! questionnaire.links.menu.yml

🔗 questionnaire.module

! questionnaire.permissions.yml

! questionnaire.routing.yml

! questionnaire.services.yml

📄 README.md

questionnaire.settings_form:

path: '/admin/questionnaire/settings'

defaults:

_form: '\Drupal\questionnaire\Form\QuestionnaireSettingsForm'

_title: 'Questionnaire Settings'

requirements:

_permission: 'administer questionnaire'

options:

_admin_route: TRUE

questionnaire.questionnaire_administration:

path: '/admin/questionnaire'

defaults:

_controller: '\Drupal\system\Controller\SystemController::systemAdminMenuBlockPage'

_title: 'Questionnaire administration'

requirements:

_permission: 'administer questionnaire'

questionnaire.display:

path: /questionnaire/{assn_id}

defaults:

_controller: '\Drupal\questionnaire\Controller\QuestionnaireDisplayController::content'

requirements:

_permission: 'access content'

options:

no_cache: TRUE

questionnaire.display_form: Jay Silverman, 11 months ago • Add custom modules. ...

questionnaire.page:

path: /questionnaire/page/{assn_id}/{page_id}

defaults:

_controller: '\Drupal\questionnaire\Controller\QuestionnaireDisplayController::getPage'

requirements:

_permission: 'access content'

options:

no_cache: TRUE

VueJS file structure

```
└─ questionnaire
  └─ modules
    └─ questionnaire-vue
      └─ dist
        └─ assets
          └─ JS index.js
          └─ JS index.js.map
          └─ # main.css
          └─ ★ favicon.ico
          └─ {} manifest.json
        └─ node_modules
        └─ public
        └─ src
          └─ components
            └─ > inputs
              └─ ▼ CompositeQuestion.vue
              └─ ▼ ConditionalSection.vue
              └─ ▼ ConditionalSubquestion.vue
              └─ ▼ DynamicInput.vue
              └─ ▼ Loading.vue
              └─ ▼ ModalAnswerHelp.vue
              └─ ▼ ModalHelp.vue
              └─ ▼ PageForm.vue
              └─ ▼ PageProgressIcon.vue
              └─ ▼ SideNav.vue
              └─ ▼ ValidationErrors.vue
            └─ > composables
            └─ > global
            └─ > router
            └─ > stores
          └─ ▼ App.vue
          └─ ▼ AppRouter.vue
          └─ JS main.js
          └─ .eslintrc.cjs
          └─ {} package-lock.json
          └─ {} package.json
          └─ ① README.md
          └─ tsconfig.json
          └─ vite.config.ts
```

App.vue

```
</script>

<template>
  Jay Silverman, 11 months ago • Add custom modules.
  <div class="questionnaire" v-if="!isLoadingMountElement">
    <SideNav
      :pages="initialStateStore.pages"
      :current-page="initialStateStore.currentPage"
      :association="initialStateStore.association"
    />
    <PageForm
      v-if="!isLoadingApi"
      :current-page="initialStateStore.currentPage"
      :pages="initialStateStore.pages"
      :association="initialStateStore.association"
    />
  </div>
</template>

<style lang="scss">
  @import url("https://fonts.googleapis.com/css2?family=Roboto+Condensed:wght@300;400&family=Roboto:wght@300;400;700&display=swap");
  $roboto: "Roboto", sans-serif;
  $roboto-condensed: "Roboto Condensed", sans-serif;
  $font-large: 50px;
  $font-medium: 35px;
  $font-text: 18px;

  $white: #ffffff;
  $grey: #323232;
  $turquoise: #007078;
  $button-primary: #10cfc9;
  $button-secondary: #7ff5f1;
  $error-primary: #781a25;
  $error-secondary: #f8d7db;
```

1. Download repo
 - a. `mkdir drupal_hybrid_midcamp_2025; cd ./drupal_hybrid_midcamp_2025`
 - b. `git clone https://github.com/keepersmith/drupal_midcamp_hybrid.git`
 - c. `cd ./drupal_midcamp_hybrid`
2. Follow “MidCamp 2025 Simple Instructions” in README.md

MidCamp 2025 Simple Instructions

```
* Make sure you already have ddev and npm (node v18+) installed, a docker container running, and you are in the correct project directory
* Run:
ddev config --project-type=drupal11 --docroot=web
ddev start
ddev composer install
ddev launch

* On "Get Started", you can "Skip this step"
* Give your site a name, and set an admin email address and password that you will easily remember.
* After "Setting up your site", log in with the admin email address and password that you just set.
* Go to /admin/modules, find "Find a Doctor", check the box, scroll down and click "Install"
* Go to /admin/structure/block, and click "Place block" in the "Content" area. Look for "Find a Doctor block", and click "Place block". Then un-check "Display title" and click "Save block".
* Go to /admin/config/development/performance, and turn off "Aggregate CSS files" and "Aggregate JavaScript files"
* View your homepage in a browser – the block should not appear yet
* Run these commands:
cd web/modules/custom/find_a_doctor/find_a_doctor
npm install
npm run watch

* Reload your homepage – see the "Vite + React" block displayed
* Make a non-breaking edit to web/modules/custom/find_a_doctor/find_a_doctor/src/App.tsx, save the file, and then reload your homepage.
* If nothing went wrong, hopefully you can now enjoy the sweet smell of victory!
```


Download our Presentation

John Brooks

VP of Engineering @sandstormdesign.com

<https://www.linkedin.com/in/john-brooks-9746824/>

jbrooks@sandstormdesign.com

Drupal.org: @keepersmith



sandstorm®

