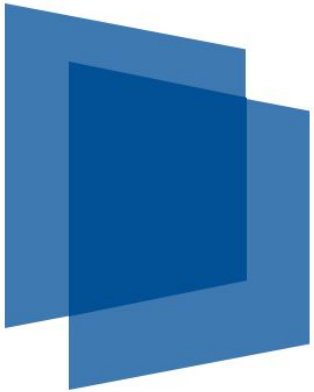

DECOUPLED DAYS 2021

presented by srijan:

Virtual Edition
July 14–15, 2021

DIAMOND SPONSORS

therefore oomph Centarro platform.sh contentful

Introducing Vuer

Background

- Drupal Outline Module
  - Used Drupal Javascript
  - Outline similar to a Drupal Book
  - Also similar to Taxonomy
    - Vocabulary is an Outline
    - Term is an Outline Entry
    - Entries organized heirarchicaly
    - Entries be stand alone piece of content or they can provide a wrapper for any Drupal Entity
- Javascript Modernization Initiative
  - Goal was to provide Admin UI functionality in a React App
  - Attempted to integrate Outline module
- Vuer / Decoupled Menus Initiative / Generic Web Components / Druxt
  - Component based approach

What does Vuer do?

- Improved Content Editing Interface for Drupal
  - drag n drop, undo/redo, graphical workflow, etc.
- Improved Book Publishing
  - Outlines can reference any Drupal entity, not just Book nodes
  - Ability to publish an outline as a document, e.g. pdf
  - Open outlines multiple Drupal servers simultaneously
- Improved Drupal Administrative UI
  - Taxonomy editing
  - Permissions
  - Workflow
  - Status Report
-

Technical architecture

Front End

- Vue / Vuex / Vuetify SPA
- Druxt
- GraphQL / Apollo

Back End

- Drupal / GraphQL
- Node.js

Vue / Vuex / Vuetify

- Vue has an easy to understand templating syntax.
- Vuex provides state management.
- Vuetify provides Material Design components.

Druxt

- Nuxt for Drupal
  - https://nuxtjs.org
- Requires a Drupal module and JS library
  - https://www.drupal.org/project/druxt
  - https://github.com/druxt
- Provides Vue components for rendering and editing Drupal entities.
  - Currently these components communicate directly with Drupal using JSON API.
  - The long term goal is to create Vue components that wrap Generic Web Components made available by the Drupal Core team.
  - Other Druxt components are provided too, e.g. site, menu, blocks, views, etc.

Backend

- Drupal
  - Requires Outline, Druxt and GraphQL modules
  - Druxt components communicate using JSON API
  - The Vue outline components communicate to Drupal using GraphQL.

- Node.js
  - Provides a place to save Vuer configuration
  - Nuxt server side rendering
  - Puppeteer screen scraping

Drupal

- Drupal Entities
  - A strength of Drupal is its robust entity support.
  - Entities are “fieldable”
  - Entities can have multiple display and edit modes.
  - Vuer can be thought of as a “Drupal entity editor” where the entities can be organized in a hierarchy.
- Drupal Core Taxonomy module similarities
  - Outline module code was a fork of the D8 Core Taxonomy module.
  - Outline is a Drupal configuration entity similar to a Taxonomy Vocabulary
  - Outline Entry is a content entity similar to Taxonomy Term
- Major differences
  - Outline Entry has only one parent
  - Outline Entry may not have custom fields
  - Outline Entry Base fields include name, content, expansion state, and entity reference
  - Link between servers
- Entity Reference Field
  - Allows an Outline Entry to point too any valid Drupal Entity
  - Outline Entry becomes a wrapper around a Drupal Entity

Development Environment

- Vue Plugin
- Lando
- VS Code
- Drupal treated as an npm package
- Provides a starting point for developing custom Vue apps that use Drupal as the back end.

Roadmap

- Content Editing
  - Integrate Druxt editing components
  - Drag/Drop, Undo/Redo, Spell Check
- Book and Taxonomy Integration
    - Editing
    - Migration
    - Eport PDF
- Auto-generate outlines / outline entries
  - e.g. Book of sale items on e-Commerce site
- Drupal Views Support
- Native App
- Static Generation

Code Walkthrough

- Drupal Outline Module
  - Based on Drupal Core Taxonomy Module where a Vocabulary is an Outline, and a Term is an Outline Entry.
- Vue App
  - Vuex
  - GraphQL/Apollo
  - Follows Nuxt project structure