

# AN OVERVIEW OF FRONT END COMPONENT INTEGRATION METHODS: DECOUPLED EDITION

**Brian Perry**

Decoupled Days – July 23, 2020

<http://bit.ly/component-int-dd>



**bounteous**

# BRIAN PERRY

- Lead Front End Dev at Bounteous
- Rocking the Chicago 'burbs
- Lover of all things components...  
...and Nintendo



**d.o: brianperry**

**twitter: bricomed**

**github: backlineint**

**nintendo: SW-4051-4946-3420**

**brianperryinteractive.com**

# bounteous



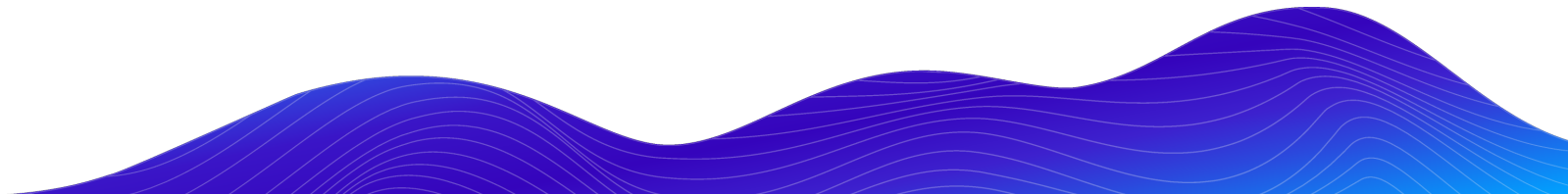
YIKES!

2020, huh?



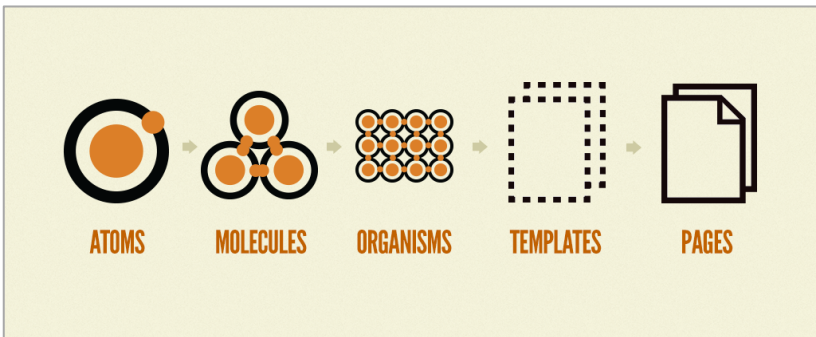


**COMPONENTS!**

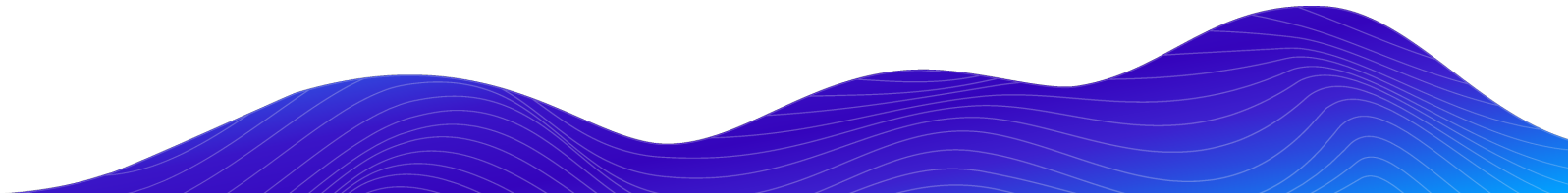


# COMPONENT BASED DEVELOPMENT

- Creating modular and re-usable elements
- Building a design system, not a series of pages
- Can use a pattern library for documentation and prototyping
  - Tools like Pattern Lab and Storybook
- Can help decouple front and back end development.



# OUR EXAMPLE WEB COMPONENT



## Containers

### Container.is-centered

Good morning. Thou hast had a good night's sleep, I hope.

### Container.is-dark

Good morning. Thou hast had a good night's sleep, I hope.

Good morning. Thou hast had a good night's sleep, I hope.

Good morning. Thou hast had a good night's sleep, I hope.



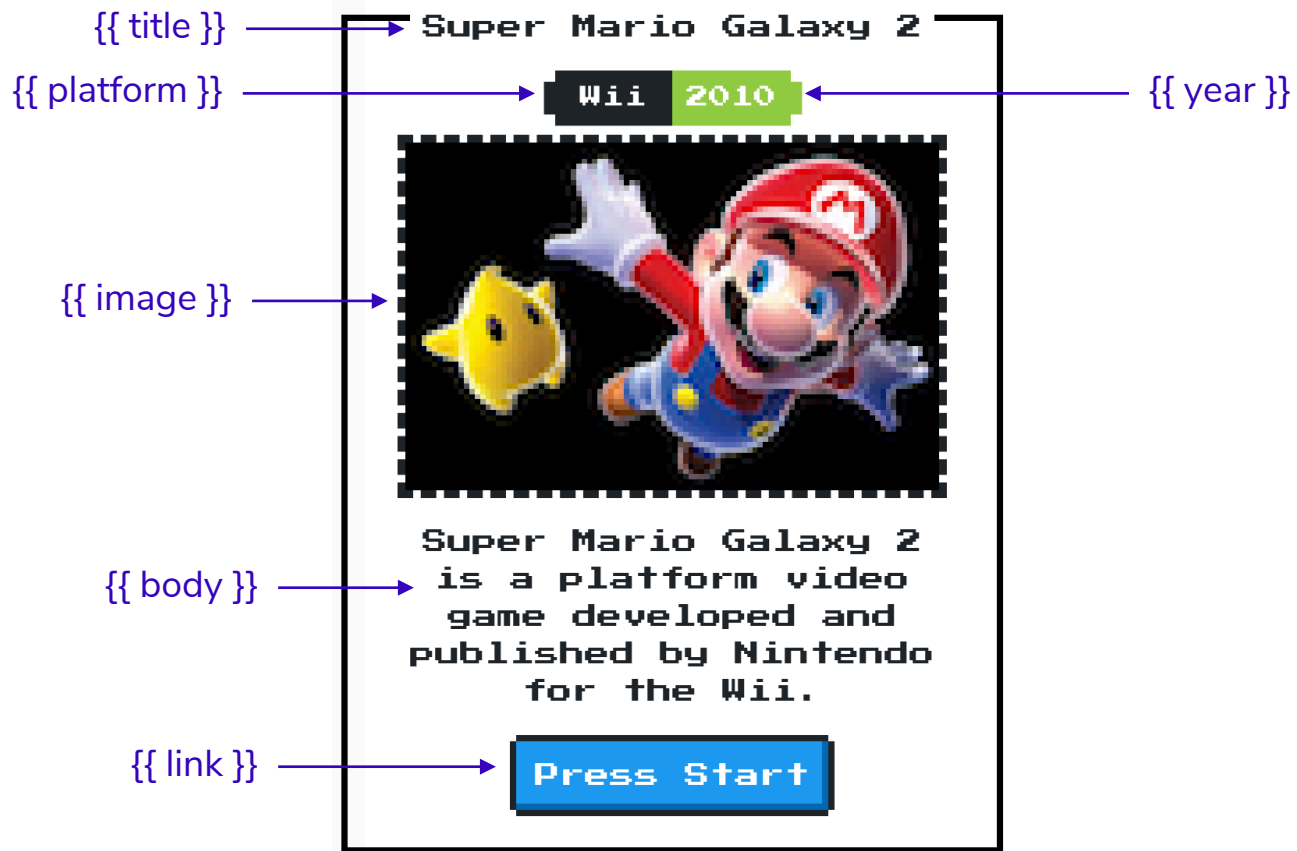
## Super Mario Galaxy 2

Wii 2010



Super Mario Galaxy 2 is a platform video game developed and published by Nintendo for the Wii.

Press Start



```
<template>
  <div class="nes-container with-title is-centered">
    <h2 v-if="title" class="title"><a :href="link">{{ title }}</a></h2>
    <div v-if="platform || year" class="nes-badge is-splited">
      <span class="is-dark">{{ platform }}</span>
      <span class="is-success">{{ year }}</span>
    </div>
    
    <p v-if="body">{{ body }}</p>
    <a v-if="link" class="nes-btn is-primary" :href="link">Press Start</a>
  </div>
</template>

<script>
export default {
  props: [
    'title',
    'link',
    'platform',
    'year',
    'image',
    'body'
  ]
}
</script>

<style src='nes.css/css/nes.min.css'></style>

<style>
  @import url('https://fonts.googleapis.com/css?family=Press+Start+2P');
</style>
```

## Super Mario Galaxy 2

Wii 2010



Super Mario Galaxy 2  
is a platform video  
game developed and  
published by Nintendo  
for the Wii.

Press Start

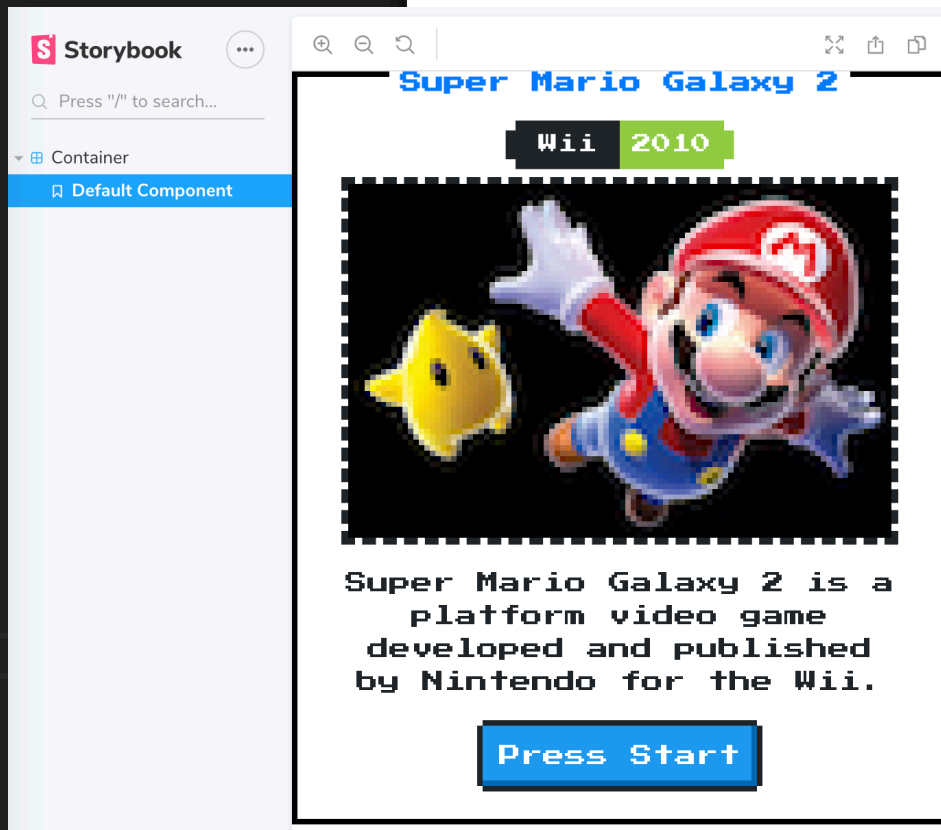
You, 13 minutes ago | 1 author (You)

```
import Container from './Container.vue';
import data from './Container.json';
import imageFile from '../assets/mario.jpg';
```

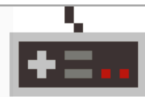
```
export default { title: 'Container' };
```

```
export const defaultComponent = () => ({
  components: { Container },
  template:
    `<Container
      title="${data.title}"
      platform="${data.platform}"
      year="${data.year}"
      image="${imageFile}"
      body="${data.body}"
      link="${data.link}"
    />`
});
```

You, 13 minutes



# GRID LAYOUT

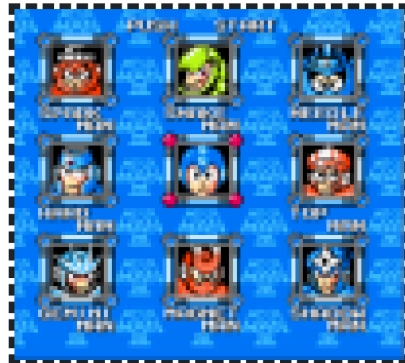


Component Integration Sandbox

## Games

### Mega Man 3

NES 1990



Mega Man 3 is an action-platform video game developed and published by Capcom.

Press Start

### Metroid Prime

GC 2002



Metroid Prime is the fifth main installment in the Metroid series, and the first Metroid game played from the first-person perspective.

Press Start

### Super Mario Galaxy 2

Wii 2010



Super Mario Galaxy 2 is a platform video game developed and published by Nintendo for the Wii.

Press Start



You, a minute ago | 1 author (You)

```
<template>
```

```
  <div class="grid">
```

```
    <slot></slot>
```

```
  </div>
```

```
</template>
```

```
<style scoped>
```

```
.grid { You, a minute ago • Added grid component
```

```
  display: grid;
```

```
  grid-column-gap: 2rem;
```

```
  grid-row-gap: 2rem;
```

```
}
```

```
@media screen and (min-width: 768px) {
```

```
  .grid {
```

```
    grid-template-columns: repeat(2, 1fr);
```

```
  }
```

```
}
```

```
@media screen and (min-width: 1024px) {
```

```
  .grid {
```

```
    grid-template-columns: repeat(3, 1fr);
```

```
  }
```

```
}
```

```
</style>
```



Component Integration Sandbox

## Games

### Mega Man 3

NES 1990



Mega Man 3 is an action-platform video game developed and published by Capcom.

Press Start

### Metroid Prime

GC 2002



Metroid Prime is the fifth main installment in the Metroid series, and the first Metroid game played from the first-person perspective.

Press Start

### Super Mario Galaxy 2

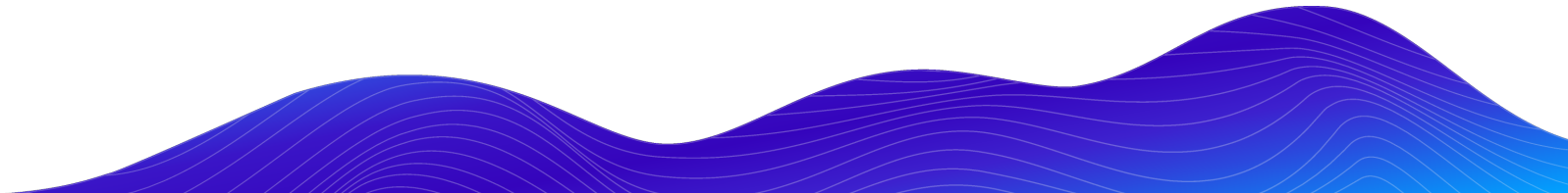
Wii 2010



Super Mario Galaxy 2 is a platform video game developed and published by Nintendo for the Wii.

Press Start

# **(TWIG) COMPONENTS IN DRUPAL**



# TRADITIONAL CMS COMPONENTS

## 3 main integration related questions:



Where do my components live?



How am I providing data to my components?



How is Drupal made aware of my components?

# TRADITIONAL CMS COMPONENTS

## 3 main integration related questions:



Where do my components live?



How am I providing data to my components?



How is Drupal made aware of my components?

# WHERE DO MY COMPONENTS LIVE?

## Traditional CMS

### Standard Drupal Components

- Live in the default template directory
- May not require any additional effort to get data to display

### Integrated Drupal Components

- Live somewhere other than the default templates directory
- Require some additional effort to get data to display

# WHERE ARE MY COMPONENTS RENDERED?

## The Decoupled Analog

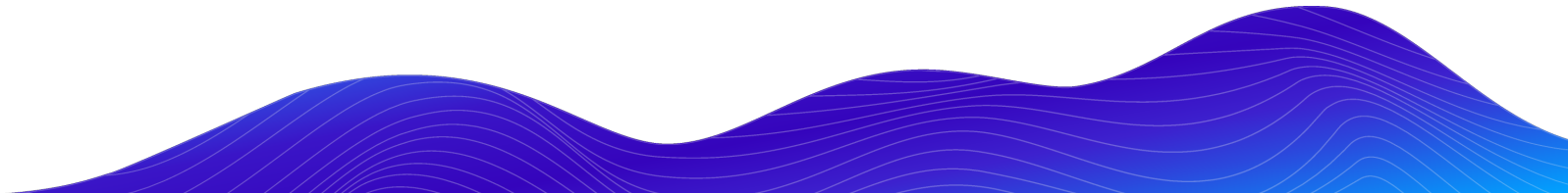
### Progressive Decoupling

- Drupal handles the initial page request and yields some portion of the page.
- Drupal has some level of visibility into your components.

### Full Decoupling

- Drupal only provides data.
- Drupal may know nothing about how the consumer will render.

# PROGRESSIVE DECOUPLING



# DRUPAL CONTENT





# Container Block Wrapper Component

- **An extremely contrived example.**
- **Could just use Drupal to render this...**

ContainerBlock.vue X

vue-nest > src > components > container > ContainerBlock.vue > {} "ContainerBlock.vue" > script > mounted

You, a few seconds ago | 1 author (You)

```
1 <template>
2   <Container
3     :title=title :platform=platform :year=year :body=body :image=image link="#"
4   />
5 </template>
6
7 <script>
8   import Container from './Container';
9
10  export default {
11    data () {
12      return {
13        title: null, platform: null, year: null, image: null, body: null
14      }
15    },
16    props: {
17      uuid: String
18    },
19    mounted () {
20      fetch(`http://decoupled-component-sandbox.lndo.site/jsonapi/node/game/${this.uuid}?include=field_image`)
21        .then(response => response.json())
22        .then(data => {
23          this.title = data.data.attributes.title
24          this.platform = data.data.attributes.field_platform
25          this.year = data.data.attributes.field_year.toString()
26          this.body = data.data.attributes.body.processed
27          this.image = `http://decoupled-component-sandbox.lndo.site${data.included[0].attributes.uri.url}`
28        });
29    },
30    components: {
31      Container
32    }
33  }
34 </script>
```

# RENDERING IN A DRUPAL BLOCK

Roll your own approach

- More work than you'd expect!
- Unfriendly to a JS Developer

# So... Much... Drupal...

- **Create a custom block type**
- **And a library to load our assets**
- **Create a theme hook**

```
! decoupled_block.libraries.yml drupal/web/modules/custom/decoupled_block/decoupled_block_libraries.yml
You, 12 hours ago | 1 author (You)
nes-container:
  version: 1.x
  css:
    theme:
      https://fonts.googleapis.com/css?family=Press+Start+2P: { type: external }
  js:
    https://unpkg.com/vue: { type: external }
    js/nes-container.min.js: {}
You, 12 hours ago • Created custom decoupled block
```

## Manage fields ☆

[Edit](#)[Manage fields](#)[Manage form display](#)[Manage display](#)

[Home](#) » [Administration](#) » [Structure](#) » [Block layout](#) » [Custom block library](#) » [Edit Decoupled Block](#)

[+ Add field](#)

LABEL	MACHINE NAME	FIELD TYPE	OPERATIONS
Game	field_game	Entity reference	<a href="#">Edit</a> ▾

```
decoupled_block.module drupal/web/modules/custom/decoupled_block/decoupled_block.module
/**
 * Implements hook_theme().
 */
function decoupled_block_theme($existing, $type, $theme, $path) {
  return array(
    'block__block_content_decoupled_block' => array(
      'template' => 'block--block-content-decoupled-block',
      'preprocess functions' => ['decoupled_block_preprocess_block']
    ),
  );
}
```

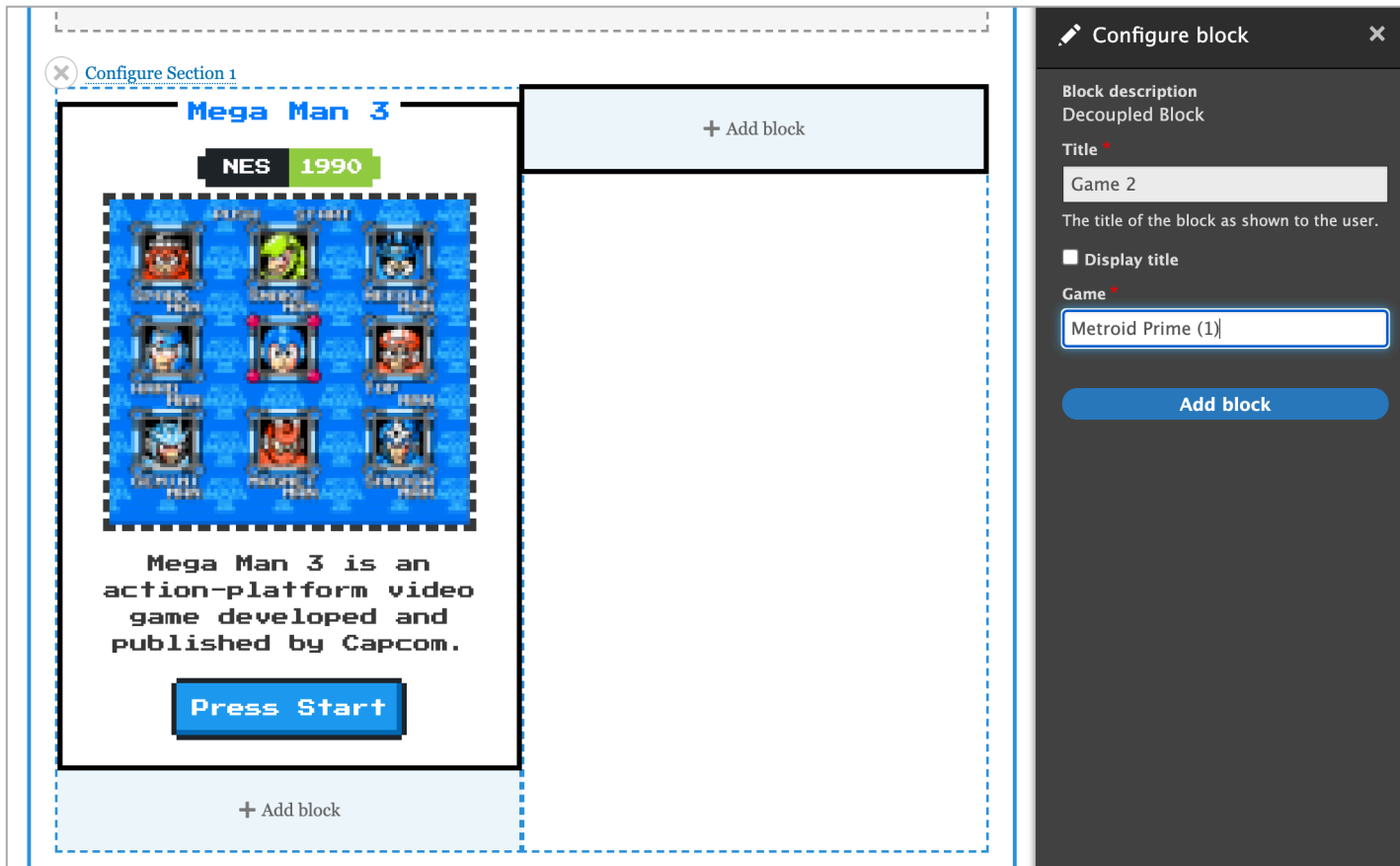
# So... Much... Drupal...

- **Preprocess the block**
- **Define the template**
- **Take a PHP nap?**

```
decoupled_block.module drupal/web/modules/custom/decoupled_block/decoupled_bloc
/**
 * Implements hook_preprocess_block().
 */
function decoupled_block_preprocess_block(&$variables) {
  $content = $variables['elements']['content'];
  if (isset($content['#block_content']) && $content['#block_content'] instanceof Blo
    if ($content['#block_content']->bundle() == 'decoupled_block') {
      $variables['uid'] = $content['#block_content']->field_game->entity->uid();
      $variables['#attached']['library'][] = 'decoupled_block/nes-container';
    }
  }
}
```

```
block--block-content-decoupled-block.html.twig drupal/web/modules/custom/decoup
*
* @see template_preprocess_block()
*
* @ingroup themeable
*/
#}
{% set component_attributes = create_attribute() %}
  You, 12 hours ago • Created custom decoupled block example
<nes-container{{ component_attributes.setAttribute('uid', uid) }}></nes-container>
```

## BUT HEY, IT WORKS



# Progressively Decoupled Blocks

- **Convention for exposing JS powered blocks to Drupal**
- **Components defined in info.yml file**
- **Simplifies framework integration**

## Decoupled Blocks

[View](#) [Version control](#) [View history](#) [Automated testing](#)

By [mrjmd](#) on 7 February 2016, updated 27 October 2016



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

This module is a javascript-framework-agnostic, progressive decoupling tool to allow custom blocks to be written by front end developers in the javascript framework of their choice, without needing to know any Drupal API's.

It keeps individual components encapsulated in their own directories containing all the css, js, and template assets necessary for them to work, and using an info.yml file to declare these components and their framework dependencies to Drupal.

It works! As of right now the Angular 2 implementation is by far the most built out, because that's where a lot of my time is being spent at my actual job. There is also a really basic React implementation, and a "hello world" example component is provided.

This is Alpha software so it is still mainly intended for the adventurous developer at this point, but you can download this module, turn on the Angular 2 or React sub-modules and place a framework-driven block on a page.

I need help! Right now the biggest needs I have are for people who:

- Know Drupal 8 really well and want to improve my D8 architecture.
- Know a javascript framework really well and want to help get it set up for use with this module. As I said above, the Angular 2 build is my main focus but I'd love to find people who'd like to help in building out the React / Ember / whatever else versions as well.

If you are interested in getting involved, let me know!

Join our weekly meetings every Monday at 4pm Eastern time here:  
[https://hangouts.google.com/hangouts/\\_/yourdrupaldev.com/pdb](https://hangouts.google.com/hangouts/_/yourdrupaldev.com/pdb)

Also a shout out to [Jason Smith](#), who dreamt up and architected the original version.

See also:

<https://github.com/mrjmd/pdb> (Github repo)  
<https://github.com/mark-casias/decoupled-blocks-drupal-vm> (Drupal VM for project quickstart)

★ Star 36 Followed

## Maintainers for Decoupled Blocks

[mrjmd](#) – 138 commits  
last: 3 years ago, first: 4 years ago  
[View all committers](#)  
[View commits](#)

## Issues for Decoupled Blocks

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

[Search](#)

[Advanced search](#)

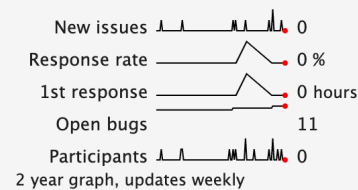
All issues

[26 open, 31 total](#)

Bug report

[11 open, 13 total](#)

Statistics



## Documentation

No documentation guides



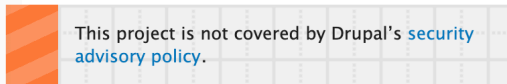
# Component

- Inspired by PDB
- Components defined by a `component.yml` file
- Creates necessary libraries
- Can define block configuration form

## Component

[View](#)[Version control](#)[View history](#)[Automated testing](#)

By [rlnorthcutt](#) on 6 May 2007, updated 14 July 2020



Adding JS components to your Drupal site just got a whole lot easier. Just combine your JS components (any type) with a ``*.component.yml`` and put it in your module or theme. Now, your component will be available in Drupal as a block – automatically!

You can also add a configuration form to your component so site builders can modify the component. This component "looks" like any other block, so it can be used just like a core block.

**JS devs don't need to know PHP or Drupal in order to integrate their components into the CMS.**

They just need to setup the ``component.yml`` file properly. The ``component.yml`` file provides the JS developer with a ton of basic configuration options. By modifying this file, you can provide static or dynamic parameters, include additional libraries, and even adjust the cache configuration. See the code comments on ``example_tabs.component.yml`` for details.

```
name: Widget
machine_name: widget
type: block
js:
  widget.js: {}
css:
  widget.css: {}
```

### Example Config

name	Leonard McCoy
greeting	Dr
active	on
interests	music,sports
favcolor	#ff2600
date	2227-01-20

```
name: Example Config
description: 'Example component showing the form config options available'
enable_field: true
js:
  example_config.js: {}
css:
  example_config.css: {}
cache:
  max-age: 0
form_configuration:
  name:
    type: textfield
    title: Name
    default_value: 'Leonard McCoy'
  greeting:
    type: select
    title: 'Greeting'
    options:
      other: 'Some other title'
      Dr: 'Dr'
      Mr: 'Mr'
      Ms: 'Ms'
      Mrs: 'Mrs'
    default_value: 'Dr'
  active:
    type: radio
    title: 'Active'
  interests:
    type: checkboxes
    title: 'Interests'
    options:
      music: 'Music'
      art: 'Art'
      sports: 'Sports'
```

[★ Unstar](#)

3

[✉ Followed](#)

## Maintainers for Component

[rlnorthcutt](#) – 21 commits

last: 1 week ago, first: 9 months ago

[Crell](#) – 6 commits

last: 10 years ago, first: 13 years ago

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

## Issues for Component

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

[Advanced search](#)

All issues

1 open, 19 total

Bug report

1 open, 6 total

Statistics

New issues 1

Open bugs 1

Participants 2

2 year graph, updates weekly

## Documentation

No documentation guides

# USING COMPONENT MODULE

```
! nes_container.component.yml drupal/web/modules/custom/container_component/components/nes-container/nes_container.component.yml

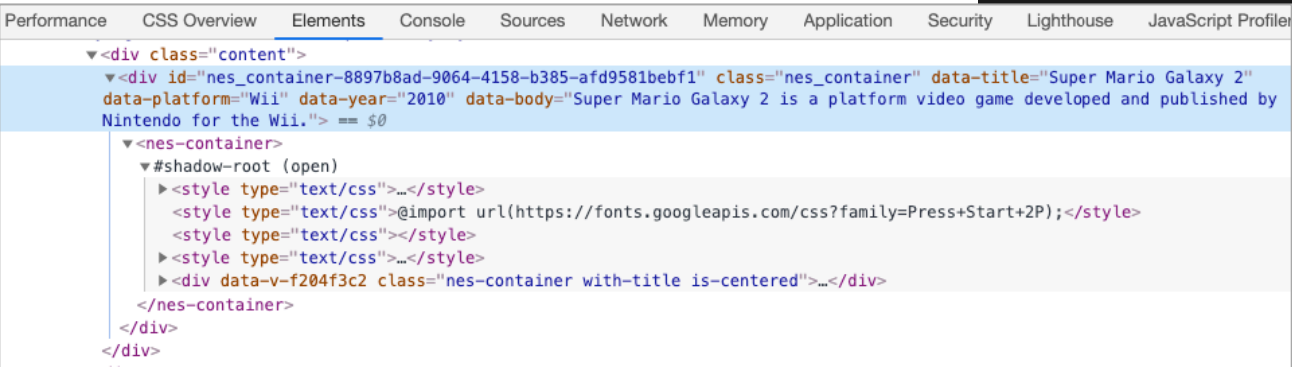
name: NES Container
description: 'Example component showing the form config options available'
enable_field: true
js:
  https://unpkg.com/vue: { type: external }
  nes-container.min.js: {}
cache:
  max-age: 60
form_configuration:
  title:
    type: textfield
    title: 'Title'
    default_value: 'Super Mario Galaxy 2'
  platform:
    type: textfield
    title: 'Platform'
    default_value: 'Wii'
  year:
    type: textfield
    title: 'Year'
    default_value: '2010'
  body:
    type: textfield
    title: 'Body '
    default_value: 'Super Mario Galaxy 2 is a platform video game developed and published by Nintendo fo
```

```

  ✓ custom
    ✓ container_component
      ✓ components / nes-container
        <> index.htm
        ! nes_container.component.yml
        JS nes-container.min.js
        ! container_component.info.yml
```



# USING COMPONENT MODULE



ContainerBlock.vue vue-nest/src/components/container/ContainerBlock.vue/ { } "Container

```
<script>
import Container from './Container';
```

```
    uid: String
  },
  mounted () {
    const drupalProvider = document.querySelector(".nes_container");
    const config = drupalProvider.dataset;
    this.title = config.title;
    this.platform = config.platform;
    this.year = config.year;
    this.body = config.body;
  },
  components: {
    Container
  }
}
</script>
```

# USING COMPONENT MODULE

Configure block

Block description: NES Container

Title \*

NES Container

Machine name: nescontainer [Edit]

☐ Display title

COMPONENT SETTINGS

Title

Super Mario Galaxy 2

Platform

Wii

Year

2010

Body

Super Mario Galaxy 2 is a platform video game developed and published by Ni

Visibility

Content types

Not restricted

Pages

Not restricted

Roles

Not restricted

Content types

☐ Article

☐ Game

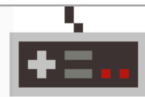
☐ Basic page

Region \*

Content

Save block

# GRID LAYOUT

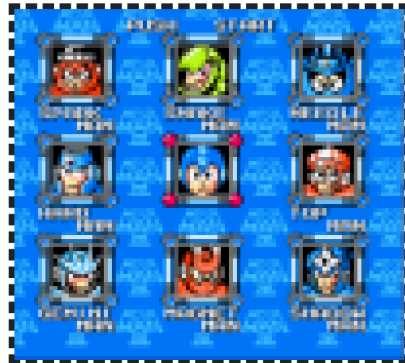


Component Integration Sandbox

## Games

### Mega Man 3

NES 1990



Mega Man 3 is an action-platform video game developed and published by Capcom.

Press Start

### Metroid Prime

GC 2002



Metroid Prime is the fifth main installment in the Metroid series, and the first Metroid game played from the first-person perspective.

Press Start

### Super Mario Galaxy 2

Wii 2010



Super Mario Galaxy 2 is a platform video game developed and published by Nintendo for the Wii.

Press Start

# Decoupled Pages

- Easily define a route that can be taken over by your JS app's routing
- I've written custom code for this repeatedly

## Decoupled Pages

[View](#)[Version control](#)[View history](#)[Automated testing](#)

By [gabesullice](#) on 15 January 2020, updated 8 April 2020

The Decoupled Pages module provides a quick and simple way to define new Drupal routes which can be overtaken by a single page application, often written in Javascript using React, Vue.js, or Ember (to name a few common frameworks).



This module doesn't have a stable release yet. You can help change that by sharing your experience with the module [in Slack](#) or [in the issue queue](#).

### How to create a decoupled page

#### 1. Defining a route

To define a new decoupled page, you define a new route in your module's [routing file](#). The [basic structure of your route definition](#) remains unchanged. You can add route requirements, route defaults and route options pretty much as normal. The only caveat is that **you must not define a `_controller` or `_form` route requirement**.

Instead, you'll define a `_decoupled_page_main` route requirement. The value of this key is the name of an asset library containing your single page application and its dependencies. It should take the form `your_module/name`. You can define this asset library in a [your\\_module.libraries.yml](#) file, but any module can provide an asset library.

Here is an example of a decoupled page route definition:

```
your_module_name.foo:
  path: /some/path/of/your-choosing
  defaults:
    _decoupled_page_main: decoupled_pages/route_test
  requirements:
    _access: 'TRUE'
```

Once your route definition has been created, a new route will be available after you've rebuilt Drupal's cache (make sure to enable your module too 🤗).

[Unstar](#)

12

[Followed](#)

### Maintainers for Decoupled Pages

[gabesullice](#) – 12 commits  
last: 2 months ago, first: 6 months ago

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

### Issues for Decoupled Pages

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

[Search](#)[Advanced search](#)

All issues

0 open, 4 total

Bug report

0 open, 2 total

Statistics

New issues 

Open bugs 

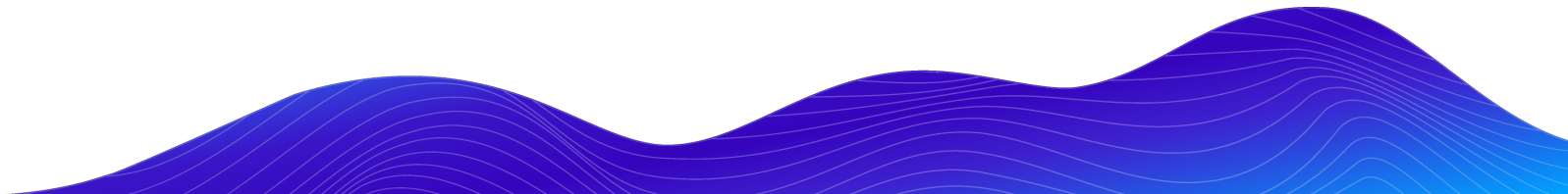
Participants 

2 year graph, updates weekly

### Documentation

No documentation guides

# FULL DECOUPLING



# TRADITIONAL CMS COMPONENTS

## 3 main integration related questions:



Where do my components live?



How am I providing data to my components?



How is Drupal made aware of my components?

# INTEGRATION APPROACHES

## Traditional CMS Components

### Mapping Data In Code

- Includes:
  - Mapping in Twig templates
  - Preprocessing

### Mapping Data In Admin UI

- Includes:
  - UI Patterns
  - Layouts

# INTEGRATION APPROACHES

## Traditional CMS Components

### Mapping Data In Code

- Includes:
  - Mapping in Twig templates
  - Preprocessing

### Mapping Data In Admin UI

- Includes:
  - UI Patterns
  - Layouts



# INTEGRATION APPROACHES

## The Decoupled Analog

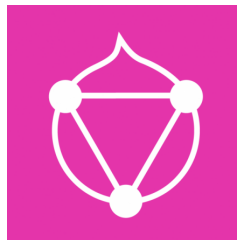
### JSON:API

- In core!
- Growing contrib ecosystem
- Multiple endpoints

The logo for JSON:API, consisting of the text "{ json:api }" in a black, monospaced font, enclosed within a light gray rectangular box.

### GraphQL

- Popular in React ecosystem
- ... but also expanding outside of React.
- Combine data from multiple sources
- Schema customization



# NO HORSE IN THIS RACE

- If your team is inclined to use GraphQL, you probably already know
- Our teams tend to use JSON:API
  - Shape of API something majority of devs are familiar with
  - Did I mention it was in Core?
  - GraphQL adoption limited / not consistent across practices.

# JSON:API ECOSYSTEM

Continues to grow

- JSON:API Extras
  - Enable/disable resources, configure field enhancers, etc.
- JSON:API Include
  - Simplify access to relationship data
- JSON:API Search API
- JSON:API Explorer
- JSON:API Cross Bundles

Something I've struggled with:

What is the right way to customize and extend JSON:API  
responses?

---

**(Hopin chat – feel free to school me on  
this one 😊)**

# SMALLER SCALE CUSTOMIZATIONS

Supported patterns:

- Create a computed field
- Write a normalizer at the data type level
- Create a field enhancer plugin using JSON:API extras

(From: <https://www.lullabot.com/articles/jsonapi-2>)

# WHAT ABOUT A COMPLETELY CUSTOM ENDPOINT?

My use case: proxying a non-Drupal service

In early days we rolled our own with json-api-php

- Solid implementation of spec
- Worry about long term maintenance challenges

Now we have some more Drupal-y options:

- JSON:API Resources
  - Seems like the most 'official' way
- API Proxy
  - Tailored to this proxy service use case

# TRADITIONAL CMS COMPONENTS

## 3 main integration related questions:



Where do my components live?



How am I providing data to my components?



How is Drupal made aware of my components?

# COMPONENT DEFINITION APPROACHES

## Traditional CMS Components

### Manual Definition

- Define component in code so that Drupal becomes aware of it.
- Likely requires some amount of duplication between Drupal and component library

### Automatic Discovery

- Drupal module automatically discovers components from component library and makes them available to Drupal.
- Emerging/experimental concept.



# AUTOMATIC DISCOVERY

Decoupled Analog - Closest thing is probably the promise of Gatsby



## Source data from anywhere

[Learn more](#)

Gatsby's rich data plugin ecosystem lets you build sites with the data you want—pull from CMSs, SaaS services, APIs, databases, your file system & more—directly into your pages using GraphQL.



+1k plugins





stripe



# GATSBY THEME CONCEPTS

## Experiment – single purpose Gatsby theme and companion Drupal module

 Community Plugin  View plugin on GitHub See starters using this

### Gatsby Theme Drupal Markdown Post

Single purpose theme that displays Markdown post data from Drupal in Gatsby with limited configuration. Uses the [gatsby-remark-drupal](#) plugin for markdown and image preprocessing.

Requires enabling the [Gatsby Markdown Post](#) module on your Drupal site.

#### Install

```
npm install --save gatsby-theme-drupal-markdown-post gatsby-source-drupal
```

This theme has [gatsby-source-drupal](#) as a peer dependency - if you aren't already using that plugin you'll need to install and configure it.

### Gatsby Markdown Post

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

By [brianperry](#) on 4 December 2019, updated 5 December 2019



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

Creates a Gatsby Markdown Post content type that can be used with the [Drupal Markdown Post Gatsby theme](#) without any additional Drupal configuration. The related Gatsby theme uses the [gatsby-remark-drupal](#) plugin for markdown and image preprocessing.

Want to try it out?

In Drupal:

- Require this module using composer
- Enable the module on your site
- Add Gatsby Markdown Post content

In Gatsby:

- Install the [Drupal Markdown Post Gatsby theme](#) and the peer dependency of [gatsby-source-drupal](#) using npm or yarn on your existing Gatsby site
- Add the [necessary configuration](#) to `gatsby-config.js`

#### Gatsby Default Starter

#### Example Drupal Markdown Post

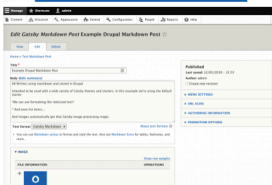
Written using markdown and stored in Drupal

Intended to be used with a wide variety of Gatsby themes and starters. In this example we're using the default starter.

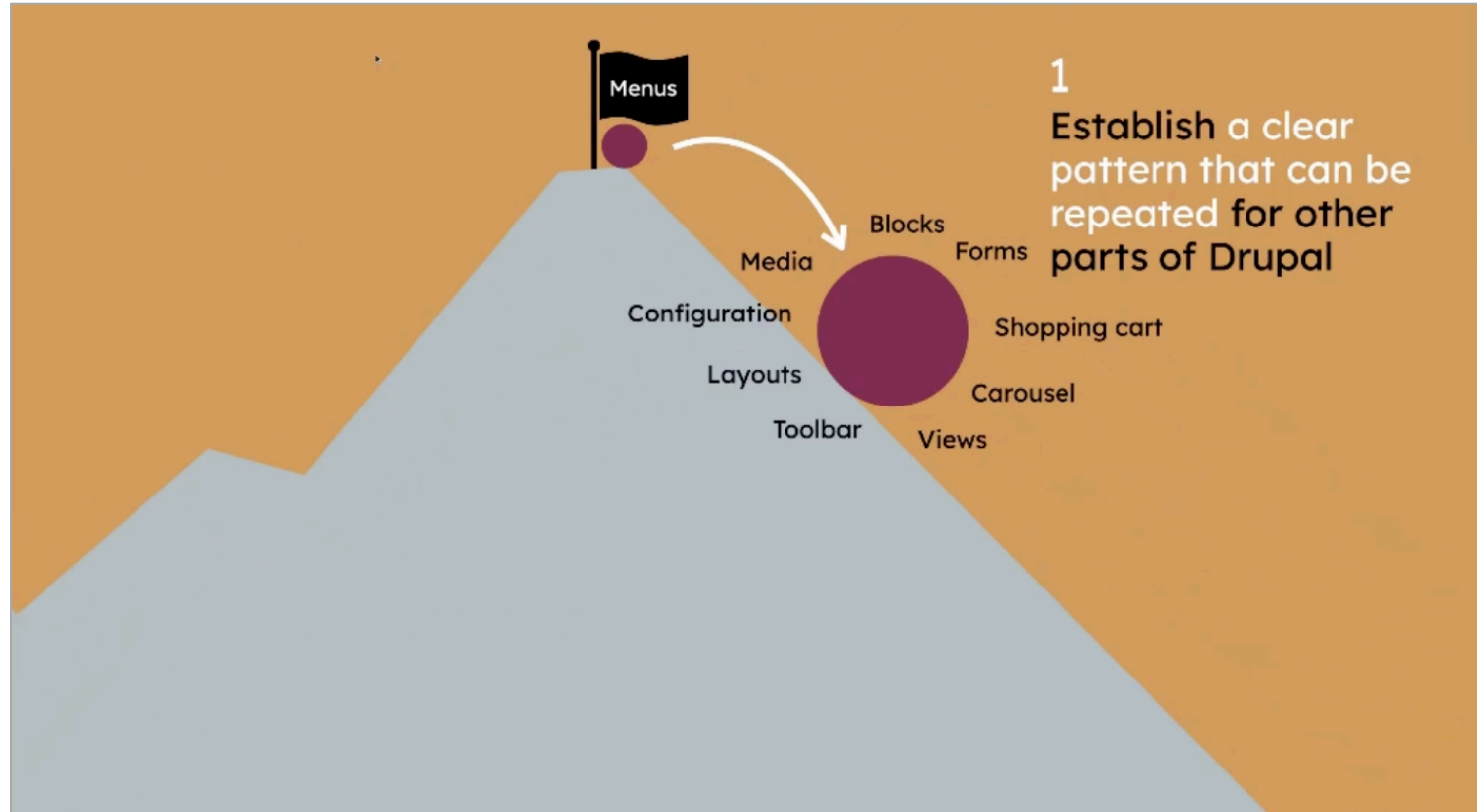
We can use formatting like *italicized text*

- And even list items...

And images automatically get that Gatsby image processing magic.



# THE FUTURE...



# I DIDN'T FINISH MY HOMEWORK

The screenshot shows the GitHub interface for the repository `backlineint / component-int-decoupled`. The repository has 1 pull request, 0 stars, and 0 forks. The main content area displays a list of files and their commit history:

File	Commit Message	Time Ago
<code>.vscode</code>	Enabled xdebug, minor style fix to container compo...	4 hours ago
<code>drupal</code>	Added additional game content	12 hours ago
<code>scripts</code>	Updated permissions for switch script	2 hours ago
<code>vue-nes</code>	Enabled xdebug, minor style fix to container compo...	4 hours ago
<code>.gitignore</code>	Initial install script	yesterday
<code>.lando.yml</code>	Enabled xdebug, minor style fix to container compo...	4 hours ago
<code>README.md</code>	Updated readme note about install and storybook	2 hours ago

Below the file list is the `README.md` content:

## Decoupled Component Integration Sandbox

Demo sandbox in support of 'An overview of Drupal component integration techniques: decoupled edition'

The right sidebar contains sections for **About** (Demo Sandbox for Drupal Component Integration Methods: Decoupled Edition), **Releases** (No releases published), **Packages** (No packages published), and **Languages** (PHP 88.4%, JavaScript 6.6%, Vue 3.9%, Other 1.1%).

# Q&A

**Brian Perry**

Lead Front End Developer

**Email:** [brian.perry@bounteous.com](mailto:brian.perry@bounteous.com)

