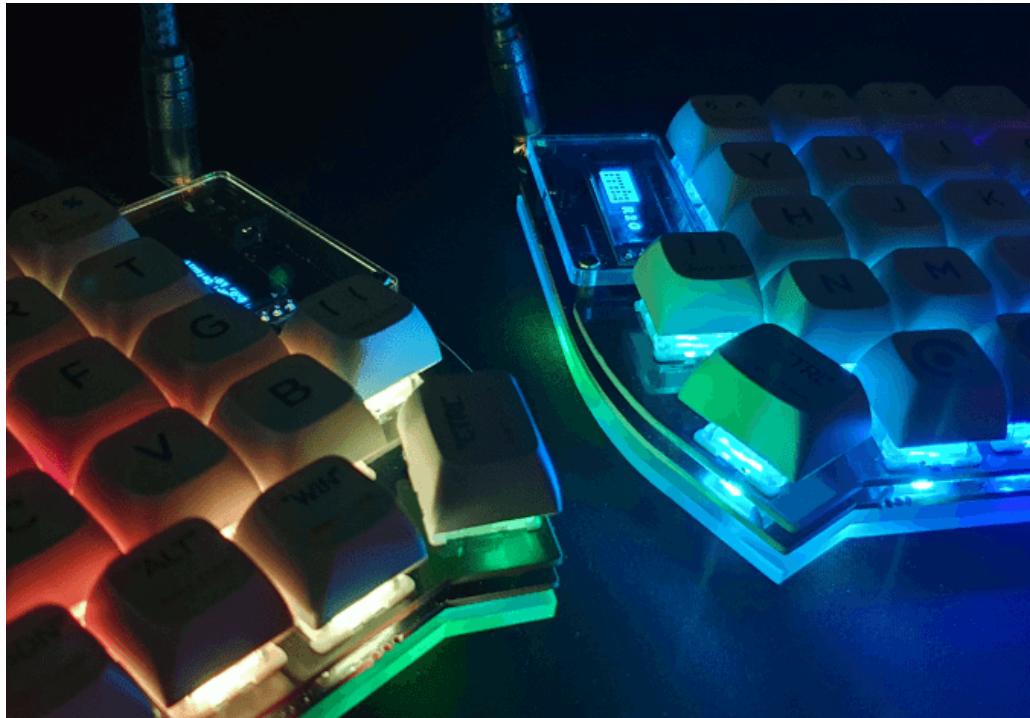


Using Storybook To Preview Single Directory Components

Philip Norton
DrupalCamp Scotland 2025

Philip Norton

- Developer at Code Enigma
- Writer at [#! code](#)
(www.hashbangcode.com)
- NWDUG co-organiser



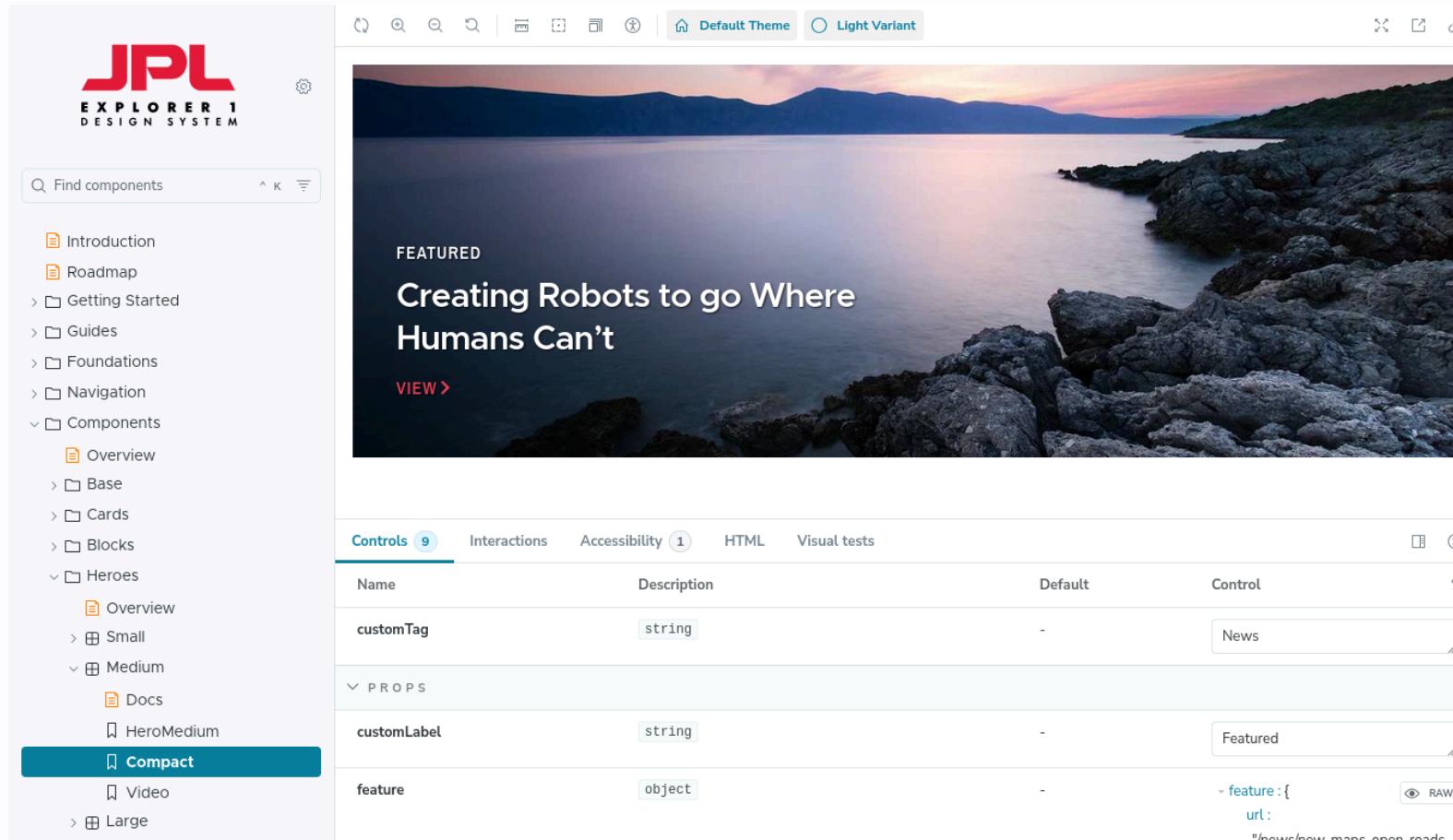
Source Code

- Talk is available at:
<https://github.com/hashbangcode/druj-storybook-talk> or via QR code.
- All code seen can be found at
<http://bit.ly/3W1bnu0>
- www.hashbangcode.com



What Is Storybook?

- Storybook is a frontend tool for creating UI components.



Why Single Directory Components?

- An important part of modern Drupal theme development.
- Can be added to themes or modules.
- Integrates with **Drupal Canvas**.

Contents

- Introduction To Single Directory Components
- Installing the Storybook Module
- Installing and running Storybook

Single Directory Components (SDC)

A *very* short introduction.



Search



Study with us

For an extraordinary future, study at one of the world's top universities.



Undergraduate Open Day

Book your place for our on-campus Undergraduate Open Day on Saturday 25 October.

Search the degree finder

Use the dropdown to select your study level.

Undergraduate

Search degree programme

Edinburgh Impact



More support for neurodivergent children



Stories of impact:
Contributing to the Sustainable



Latest news
Edinburgh first UK uni to scoop nutrition award



Autumn Festivals aglow with inspiring events

Study Global Visit Research News About Alumni Local Staff Students

THE UNIVERSITY
of EDINBURGH

Schools & departments MyEd

Search

Study with us

For an extraordinary future, study at one of the world's top universities.

Undergraduate Open Day

Book your place for our on-campus Undergraduate Open Day on Saturday 25 October.

Search the degree finder

Use the dropdown to select your study level.

Undergraduate ▾

Search degree programm

Edinburgh Impact

More support for neurodivergent children

Stories of impact:
Contributing to the Sustainable

Latest news
Edinburgh first UK uni to scoop nutrition award

Autumn Festivals aglow with inspiring events

9

SDC

- They exist in a single directory (hence the name).

```
example_component/
    example_component.component.yml
    example_component.css
    example_component.js
    example_component.twig
```

SDC - *.component.yml

- The `*.component.yml` file is used to define the structure of the component.

```
name: My Component
```

```
props:
```

```
  type: object
```

```
  properties:
```

```
    # ...
```

```
slots:
```

```
  # ...
```

SDC - *.component.yml

- `props` have defined properties with set data types.

```
props:  
  type: object  
  properties:  
    title:  
      type: ['string', 'null']  
      title: Extra classes  
    examples:  
      - extrawide, dark
```

- `{{ title }}` can now be used in the template.

SDC - *.component.yml

- `slots` are much simpler to define.

```
slots:  
  body:  
    title: "Body copy."
```
- `{{ body }}` is now available in the template.

SDC - Props vs Slots

- `props` - Properties or simple elements that the component has. Title, classes, menu depth, display options, etc.
- `slots` - Objects or more complex elements that the component can render. Body copy, images etc.

SDC - *.twig

- A normal twig template that renders the defined structure of the component.

Slot in a *.component.yml file:

```
slots:  
  title:  
    title: "Title"
```

Twig template:

```
{{ title }}
```

SDC - *.twig

- You should probably add `slots` using the `block` Twig tag.

```
{% block title %}  
  {{ title }}  
{% endblock %}
```

- This allows you to override the value in Storybook without having to add `|raw` to your Twig template.

SDC - Other files

- Any `*.css` or `*.js` files will be automatically included as attachments.
- Template and CSS can reference files inside the SDC.

[[[add some examples of images being referenced from SDC]]]

SDC - Usage

- With the SDC created we can now add it to our Drupal template templates.

```
 {{ include('my_theme:example_component', {  
   title: label,  
   body: content.body,  
 }) }}
```

SDC - Usage

- An example of rendering an Author content type.

```
{{ include('my_theme:author', {  
    author_url: url,  
    name: label,  
    bio: content.field_author_short_bio,  
    avatar: content.field_author_avatar  
}) }}
```

Storybook Module

Storybook Module

- Require and install the Storybook module

```
composer require drupal/storybook  
drush en storybook
```

Storybook Module - Permissions

- Ensure that the correct permissions are set.

```
drush role:perm:add anonymous 'render storybook stories'
```

Storybook Module - Stories

- For each component you want to test you need a `*.stories.twig` file.
- This contains a number of examples of the SDC in use.
- Add additional markup or other components to the story.

Storybook Module - Stories

```
{% stories componentName with { title: 'Components/ComponentName' } %}

{% story default with {
  name: '1. Default',
  args: {
    some_property: 'A value'
  }
} %}
{% embed 'my_theme:componentName' %}
{% endembed %}

{% endstory %}

{% endstories %}
```

Storybook Module - Stories

- [[look at how stories are translated into Storybook previews]]

Storybook Module - Stories

- Use `include` for simple components.

`example_component.twig`

```
<div>
  <h2>{{ title }}</h2>
</div>
```

`example_component.stories.twig`

```
{% story default with {
  name: '1. Default',
  args: {
    title: 'The Title'
  }
} %}
{{ include('my_theme:example_component', {
  title
}) }}
```

Storybook Module - Stories

- Use `embed` with more complex components that have slots.

`example_component.twig`

```
<div>
  {% block body %}
    {{ body }}
  {% endblock %}
</div>
```

`example_component.stories.twig`

```
---
  {% story default with {
    name: '1. Default',
    args: {
      body: '<p>Some body copy.</p>'
    }
  } %}
  {% embed 'my_theme:example_component' %}
    {% block body %}
      {{ body|raw }}
    {% endblock %}
  {% endembed %}
---
```

Storybook Module - Stories

- Storybook doesn't understand `*.stories.twig` files.
- They need converting into `*.stories.json`.
- This Storybook module comes with a couple of Drush commands to do this.

```
drush storybook:generate-all-stories
```

Drupal Setup

- Some settings are needed to allow Storybook to connect to Drupal.

Drupal Setup - CORS Prevention

- Add these rules to `development.services.yml` file.

```
parameters:  
  storybook.development: true  
cors.config:  
  enabled: true  
  allowedHeaders: [ '*' ]  
  allowedMethods: [ '*' ]  
  allowedOrigins: [ '*' ]  
  exposedHeaders: false  
  maxAge: false  
  supportsCredentials: true
```

Drupal Setup - CORS Prevention

- Add a reference to the services file from the `settings.php` file.

```
$settings['container_yamls'][] = DRUPAL_ROOT . '/sites/development.services.yml';
```

Drupal Setup - Twig Debug

Turn on the theme development settings.

```
drush state:set twig_debug 1  
drush state:set twig_cache_disable 1  
drush state:set disable_rendered_output_cache_bins 1
```

[[[[Insert a picture of the twig debug page here]]]]

Storybook

Storybook - Install

- Install Storybook in a separate directory, away from your Drupal root.
- Eg. .

```
npm init -y
npx storybook init --type server
```

Storybook - Install

- This will give you a directory containing.

```
.storybook  
stories  
node_modules  
package.json  
package-lock.json
```

Storybook - Configure

- Edit the `.storybook/main.js` file to point Storybook at your stories.

```
"stories": [  
    ".../..../web/themes/custom/my_theme/components/**/*.{stories.@(json|yaml|yml)}"  
],
```

Running Storybook

- Storybook is a JavaScript application that can be run on its own.

```
npm run storybook
```

- This will open a browser window with the address <http://localhost:6006>.

Running Storybook - DDEV

- If you are using DDEV then can run Storybook from inside the web container, with some config.

```
web_extra_exposed_ports:  
  - name: storybook  
    container_port: 6006  
    http_port: 6007  
    https_port: 6006  
web_extra_daemons:  
  - name: node.js  
    command: "tail -F package.json > /dev/null"  
    directory: /var/www/html/tests
```

Running Storybook - DDEV

- To stop DDEV complaining you may want to add
--no-open flag the relevant line in the
package.json file to prevent the browser opening.

```
"scripts": {  
  "storybook": "storybook dev -p 6006 --no-open",  
  "build-storybook": "storybook build"  
},
```

Building Storybook

- Storybook can also be built as a static website.

```
npm run build-storybook
```

- The static site can be deployed and served as a site.

Hosting Storybook

- Use the following nginx configuration.

```
location ^~ /storybook-static {  
    add_header "Access-Control-Allow-Origin" *;  
    add_header "Access-Control-Allow-Methods" "GET, POST, OPTIONS";  
    disable_symlinks off;  
    alias /your/project/directory/tests/storybook-static;  
    index index.html;  
}
```

Workflow

- A workflow for building a component.
 - Create story for component
 - Build (or run) storybook
 - Preview story
 - Once the SDC is complete, add it to your template.

Demo

Tips

- Look for CORS errors.
- If you only see a blank screen, check Drupal logs.
- Be aware that Drupal will wrap some things with extra elements. That might throw off your positioning and margins.
- Setup a styleguide storybook.

Tips

- Understand props and slots.
- Don't abstract too much with SDC.
- Don't try to cover too much with SDC.

Alternatives

- SDC - Component library

[[[ADD SCREENSHOT OF SDC COMPONENT LIBRARY MODULE N USE HERE]]]

Alternatives

- Single Directory Components Styleguide

[[[ADD SCREENSHOT OF SDC COMPONENT
STYLEGUIDE MODUE N USE HERE]]]

Resources

- Storybook
- Storybook Module
- Drupal Documentation on SDC
- Drupal 11: Using Storybook To Preview Single Directory Components

Questions?

- Slides:
<https://github.com/hashbangcodestorybook-talk>



Thanks!

- Slides:
[https://github.com/hashbangcod
storybook-talk](https://github.com/hashbangcod/storybook-talk)

