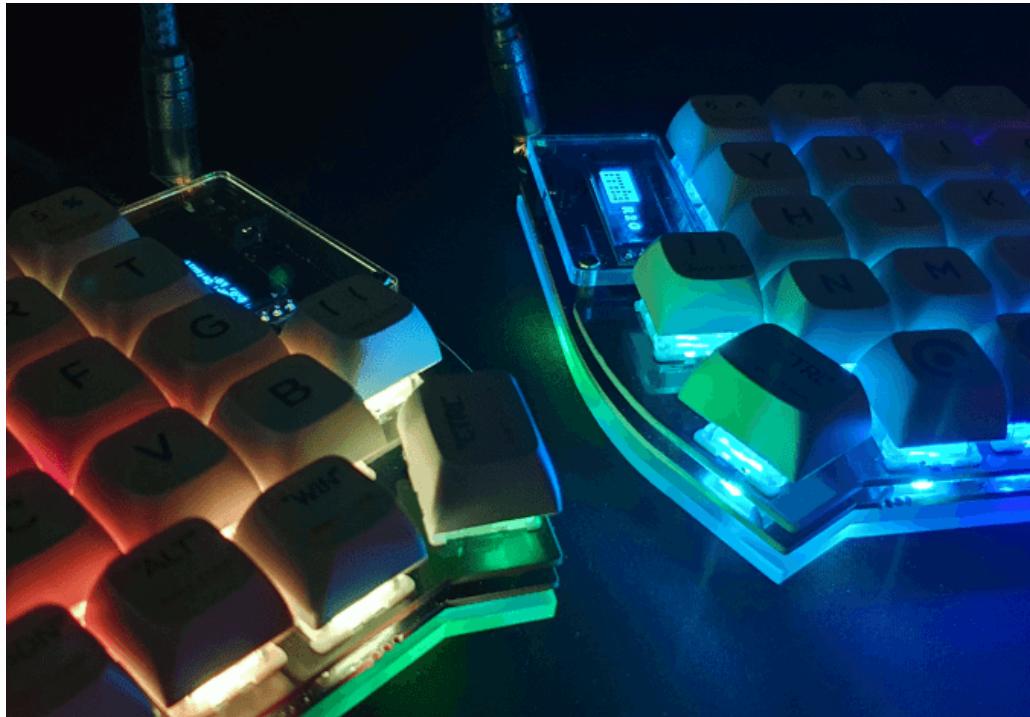


# Using Storybook To Preview Single Directory Components

Philip Norton  
DrupalCamp Scotland 2025

# Philip Norton

- Developer at Code Enigma
- Writer at [#! code](#)  
([www.hashbangcode.com](http://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](http://www.hashbangcode.com)



# What Is Storybook?

- Storybook is a frontend tool for creating UI components.

The screenshot shows the JPL Explorer 1 Design System Storybook interface. On the left, there's a sidebar with the JPL logo and a search bar labeled "Find components". Below the search bar is a navigation menu with sections like "Introduction", "Roadmap", "Getting Started", "Guides", "Foundations", "Navigation", "Components" (which is expanded to show "Overview", "Base", "Cards", "Blocks", "Heroes" (expanded to show "Overview", "Small", "Medium", "Docs", "HeroMedium"), and "Compact"). At the bottom of the sidebar, there's a button labeled "Compact". The main area features a large image of a sunset over a rocky coastline. Overlaid on the image is a "FEATURED" section with the title "Creating Robots to go Where Humans Can't" and a "VIEW >" button. Below this, there's a table titled "Controls" with 9 items. The first item is "customTag" with a "string" type, currently set to "News". The second item is "customLabel" with a "string" type, currently set to "Featured". There are also sections for "PROPS" and "feature".

Controls	9	Interactions	Accessibility	1	HTML	Visual tests
Name	Description	Default	Control			
customTag	string	-	News			
PROPS						
customLabel	string	-	Featured			
feature	object	feature	feature			

# 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.



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

Stories of impact: Contributing to the

Latest news

Autumn Festivals aglow with



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



Stories of impact: Contributing to the



Latest news



Autumn Festivals aglow with

# 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:  
    extra_classes:  
      type: ['string', 'null']  
      title: Extra classes  
    examples:  
      - extrawide, dark
```

- `{{ extra_classes }}` 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 - \*.component.yml

- Use `libraryOverrides` to inject Drupal libraries into your SDC.

```
libraryOverrides:  
  dependencies:  
    - core/drupal  
    - core/once
```

# SDC - \*.twig

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

Slot in a \*.component.yml file:

```
slots:  
  body:  
    title: Body copy  
    description: The HTML body copy of the element.
```

Twig template:

```
{{ body }}
```

# SDC - \*.twig

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

```
{% block body %}  
  {{ body }}  
{% 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.
- The libraries attached to the theme will also be included when the SDC is rendered in Storybook.

# SDC - Other files

- Template and CSS can reference files inside the SDC.

CSS:

```
background-image: url('./assets/background.png');
```

Image:

```

```

# SDC - Usage

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

```
 {{ include('my_theme:example_component', {  
   extra_classes: 'some-class',  
   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 exampleComponent with { title: 'Components/Example Component' } %}

{# Define story name and args: #}
{% story default with {
  name: '1. Default',
  args: {
    some_property: 'A value'
  }
} %}

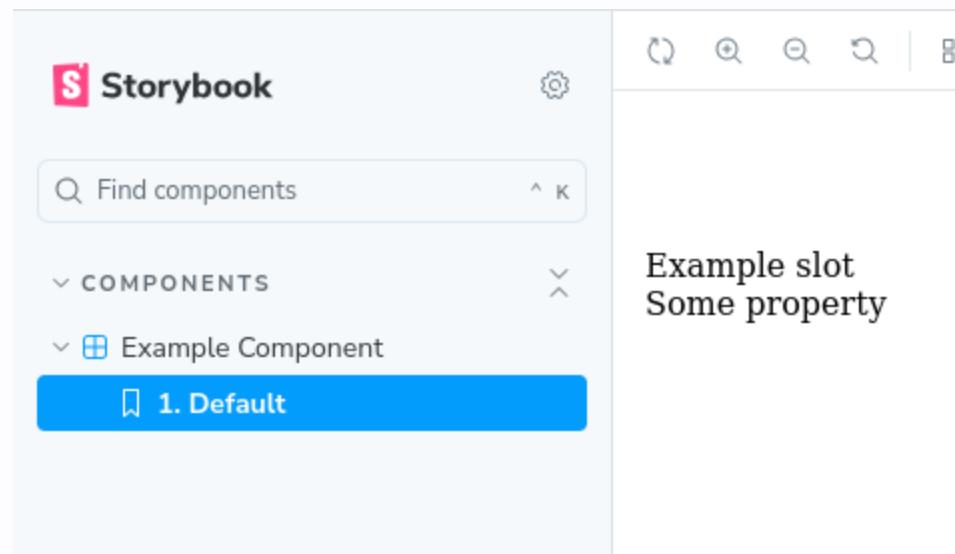
{# Embed component: #}
{% embed 'my_theme:example_component' %}
{% endembed %}

{% endstory %}

{% endstories %}
```

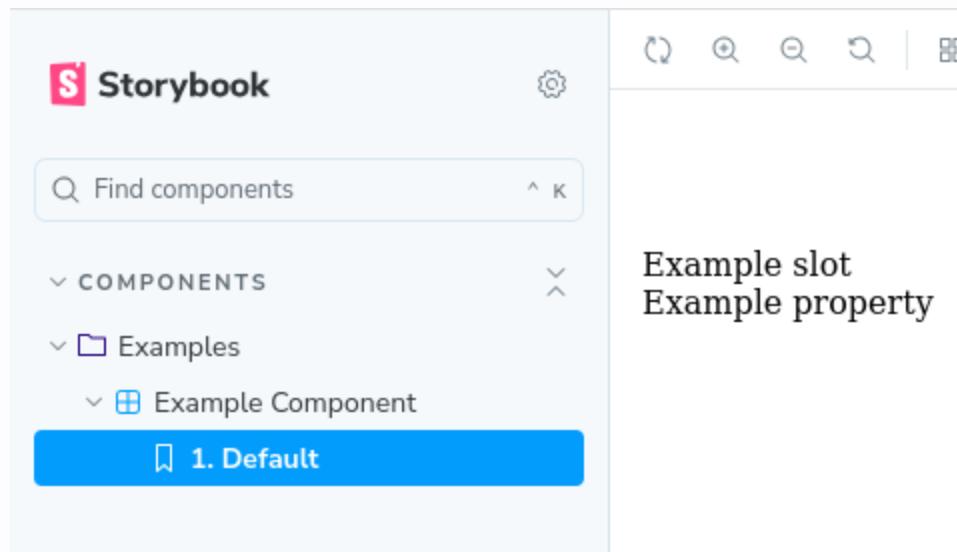
# Storybook Module - Stories

```
{% stories exampleComponent with { title: 'Components/Example Component' } %}
```



# Storybook Module - Stories

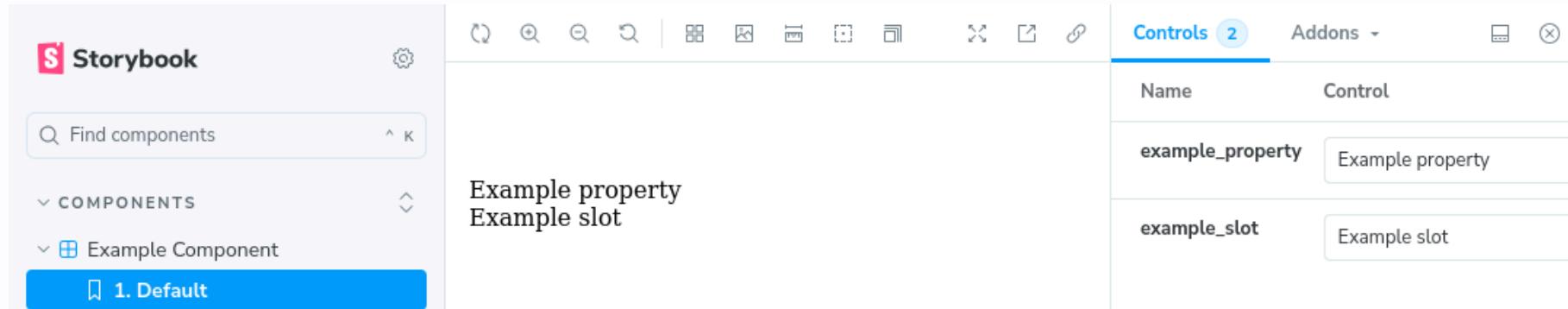
```
{% stories exampleComponent with { title: 'Components/Examples/Example Component' } %}
```



# Storybook Module - Stories

- Any args passed to the story are editable properties.

```
{% story default with {  
  name: '1. Default',  
  args: {  
    example_property: 'Example property',  
    example_slot: 'Example slot',  
  }  
} %}
```



# 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

- You can add multiple stories for your component.

```
{% story wrapped with {
  name: '2. Wrapped',
  args: {
    example_property: 'Example property',
    example_slot: '<p>Example slot with lots of content.</p>',
  }
} %}
<div style="border:1px solid red; width: 100px;">
  {% embed 'my_theme:example_component' %}
    {% block example_slot %}
      {{ example_slot|raw }}
    {% endblock %}
  {% endembed %}
</div>
{% endstory %}
```

# 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
```

# Storybook Module - Stories

- Don't commit the `*.stories.json` files, they are created for the current site.

# 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';
```

- Don't put these rules on your production instance!

# Drupal Setup - Twig Debug

- Turn on the theme development settings.

## Development settings ☆

These settings should only be enabled on development environments and never on production.

Do not cache markup

Disables render cache, dynamic page cache, and page cache.

Twig development mode

Exposes Twig development settings.

### Twig development mode

Twig debug mode

Provides Twig's dump() function for debugging, outputs template suggestions to HTML comments, and automatically recompile Twig templates after changes.

Disable Twig cache

Twig templates are not cached and are always compiled when rendered.

# Drupal Setup - Twig Debug

Turn on the theme development settings (via Drush).

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

# Storybook

# Storybook - Install

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

```
mkdir tests && cd tests  
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.
- This means that you can demo SDCs to clients before they are written into Drupal template.

# Building Storybook - Hosting

- 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;  
}
```

# Building Storybook - Hosting

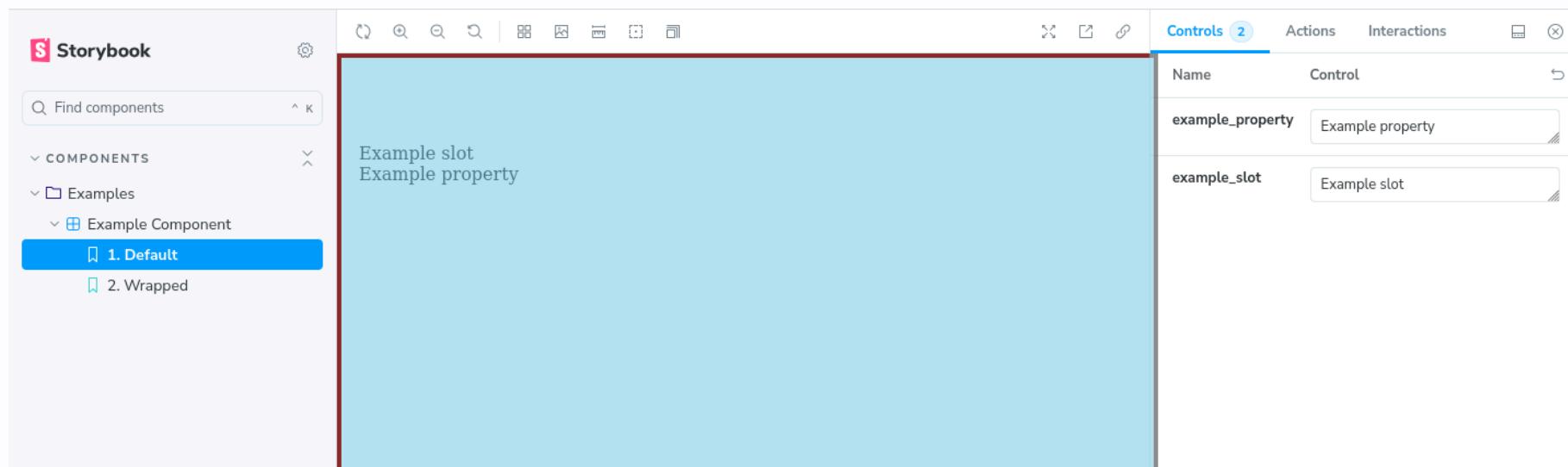
- Extra configuration is needed for external files.

```
location @rewrite {
    rewrite ^ /index.php;
}

location ~* \.(js|css|woff|woff2|ttf)$ {
    add_header Access-Control-Allow-Origin '*';
    add_header X-Content-Type-Options nosniff;
    try_files $uri @rewrite;
    expires -1;
    log_not_found off;
}
```

# Storybook - SDC

- The whole mid section here is an iframe that points to Drupal.



# 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.
- Storybook loves to cache the sidebar.
- Storybook has lots of plugins, check a few of them out.

# Tips

- Be aware that Drupal will wrap some things with extra elements. That might throw off your positioning and margins.
- Setup a styleguide storybook.
- Understand `props` and `slots`.
- Don't abstract too much with SDC.
- Don't try to cover too much with SDC.

# Alternatives

# Alternatives

- SDC - Component library

## Components Preview

Find components

**My Theme components**

Author

**Author - my\_theme**

Show code

 Phil Norton

Phil is the founder and administrator of [#! code](#) and is an IT professional working in the North West of the UK.

## Accessibility Issues

*This is not an exhaustive check, and WCAG issues may still be present. Please perform additional accessibility tests for full compliance.*

- No accessibility issues found!

# Alternatives

- Single Directory Components Styleguide

The screenshot shows a user interface for a 'Styleguide' component. The top navigation bar includes 'Manage', 'Shortcuts', 'admin', 'Announcements', and 'Edit' buttons. Below the navigation is a horizontal menu with links for 'Content', 'Structure', 'Appearance', 'Extend', 'Configuration', 'People', 'Reports', and 'Help'. On the left side, there is a dark blue sidebar with the 'Styleguide' logo at the top. Below the logo, the sidebar contains a navigation menu with the following items:

- Colors
- Text
- Ungrouped
  - Example Component
    - Example Demo
  - Ratings

The main content area displays a series of headings from H1 to H6, each followed by a short paragraph of placeholder text ('Lorem ipsum dolor sit amet').

Heading Level	Text Content
H1	This is an H1
H2	This is an H2
H3	This is an H3
H4	This is an H4
H5	This is an H5
H6	This is an H6

Below the headings, there is a paragraph of placeholder text:

This is a paragraph with lorem ipsum. [Lorem ipsum dolor sit amet](#), consectetur adipiscing elit. Nam faucibus quam ut tincidunt condimentum. Aenean vulputate velit vel erat consectetur, sed malesuada metus vulputate. Aliquam ex odio, pretium non quam ut, varius suscipit velit. Maecenas odio neque, maximus a ultricies eu, mattis sed ex. Maecenas purus felis, porttitor in tempor at, condimentum ac mi. Donec lacinia lorem quis nisi consectetur mollis. Fusce pellentesque nulla a fringilla sodales. Sed a lorem orci. Vivamus fringilla nisl at limula dismmissim faucibus.

# Alternatives

- Storybook SDC Addon
- Plugin for Storybook that allows you to build SDCs as native Storybook stories.

# 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)

