Using Paragraphs in Drupal 8

By Ivan Zugec from WebWash

Contents

Paragraphs Module for Drupal 8

Over the last couple of years edge-to-edge, or fluid looking websites, have become popular. You just have to look at a few digital agency websites and you'll notice the common pattern.

With this design, the page is broken out into separate components. A component could be something as simple as a single text field, or something complex like an image gallery. You may also see scrolling animation on these components. As a user scrolls the text could zoom in from the right and an image could fade in from the left.

How would you build this type of website in Drupal?

Let me introduce you to Paragraphs.

The module allows you to build *Paragraph types* which an editor can then add to a page using a Paragraph field.

On a decent size website, you could have a few paragraph types: one for entering in text and another to display content in a grid, or another which displays a video.

In this tutorial, you'll learn the basics of Paragraphs and we'll setup a simple paragraph type for adding basic text.

In future tutorials, we'll look at using Paragraphs to manage a dynamic sidebar and how to create an image gallery.

We have a lot to do, so let's jump right in.

Getting Started

Before we begin, download and install Paragraphs and Entity Reference Revisions.

Below are the Drush and Drupal Console commands to download and install:

Drush:

```
$ drush dl paragraphs entity_reference_revisions
$ drush en paragraphs
```

Drupal Console:

```
$ drupal module:download paragraphs
$ drupal module:download entity_reference_revisions
$ drupal module:install paragraphs
```

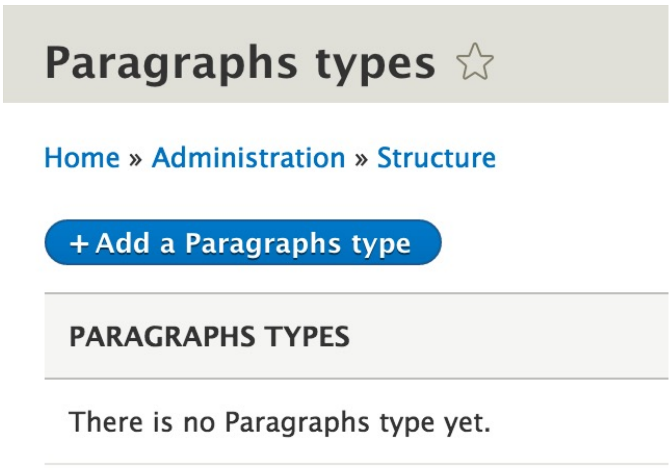
Creating Paragraph Type

When creating a *paragraph-powered* website the first thing you need to do is look at the designs and look out for common patterns.

For example, there may be a need to display an image and text side-by-side in a grid, or testimonials may need to be displayed. These two examples are perfect candidates for their own paragraph type.

For this tutorial, we'll create a basic Content paragraph type which'll have just two fields: Title (text field) and Content (textarea).

1. On the navigation bar click on Structure and "Paragraphs types".



Paragraphs types ☆

Home » Administration » Structure

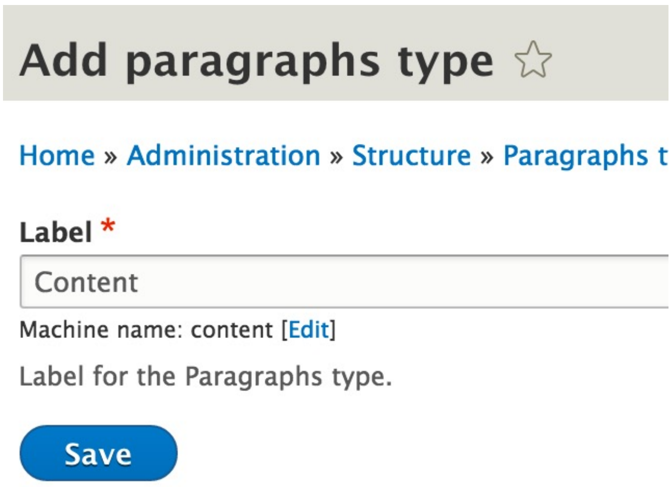
+ Add a Paragraphs type

PARAGRAPHS TYPES
There is no Paragraphs type yet.

From this page you can create new paragraph types and you'll be able to manage existing ones. Currently, there are none, so you can't see any.

2. Click on "Add a Paragraphs type".

3. Enter Content into the Label and make sure the machine name is `content`. Then click on Save.



Add paragraphs type ☆

Home » Administration » Structure » Paragraphs t

Label *

Content

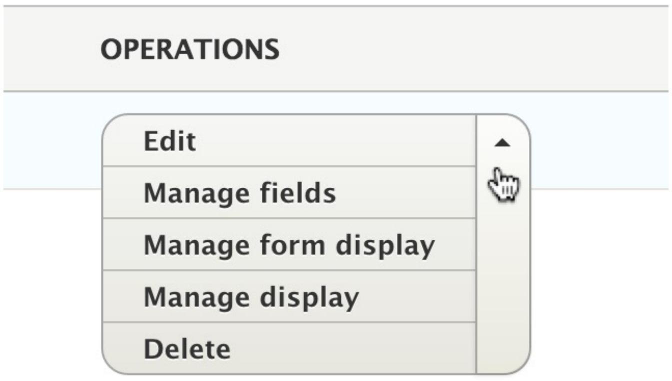
Machine name: content [Edit]

Label for the Paragraphs type.

Save

4. Now you should be back on the “Paragraphs types” page and you should see the new Content paragraph type which we just created.

If you click on the Edit button down arrow, in the Operations column, you should see a bunch of options such as “Manage fields” and “Manage displays”.

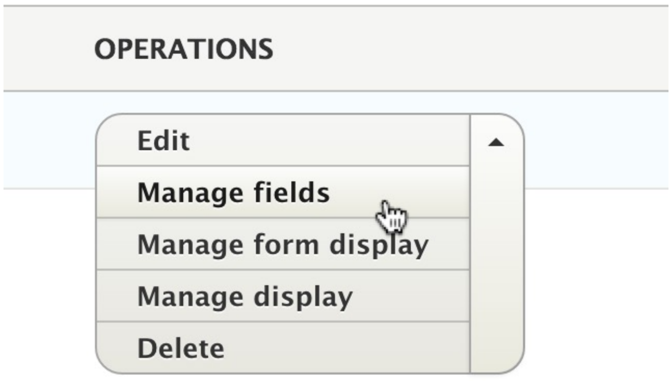


If you know how to add fields to content types or other entities than this should look familiar. Managing fields on a paragraph type is the same as on content types and other entities.

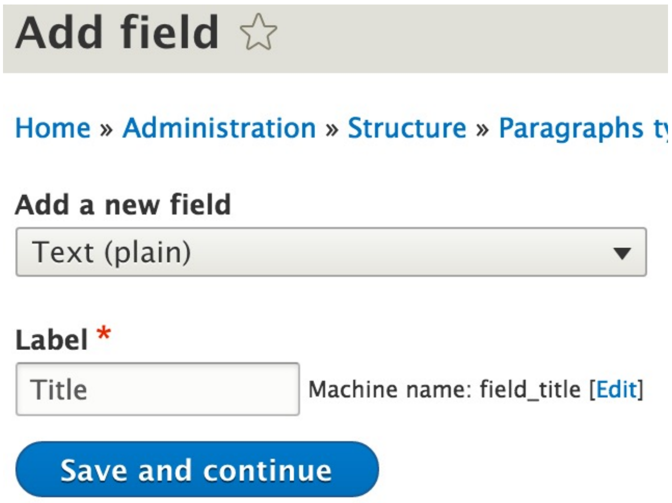
Adding Fields to Paragraph Type

I mentioned earlier that the Content paragraph type will have two fields: Title and Content. Let’s now create those fields.

1. From the Edit drop-down click on “Manage fields”.



2. Click on “Add field”, select “Text (plain)” from the “Add a new field” drop-down and enter in Title into the Label field.


Add field

Home » Administration » Structure » Paragraphs ty

Add a new field: Text (plain)

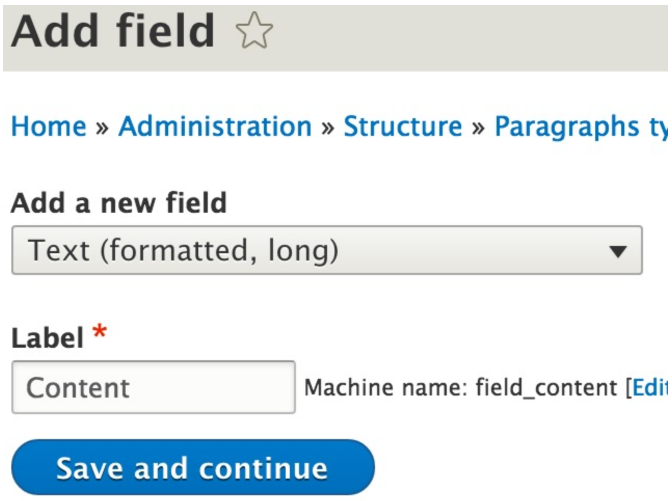
Label *: Title Machine name: field_title [Edit]

Then click on “Save and continue”.

3. Leave the next page, “Field settings” as is and click on “Save field settings”.

4. Leave the Edit page as is as well and click on “Save settings”.

5. Now, let’s create the Content field. Click on “Add field”, select “Text (formatted, long)” from the “Add a new field” drop-down and enter in Content into the Label field.


Add field

Home » Administration » Structure » Paragraphs ty

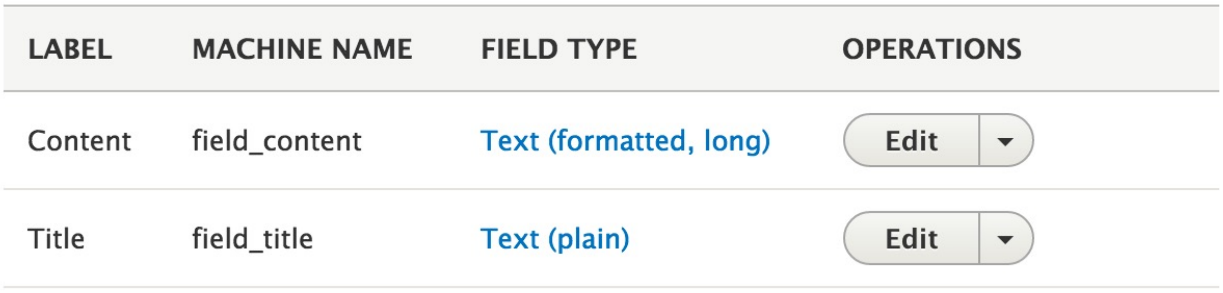
Add a new field: Text (formatted, long)

Label *: Content Machine name: field_content [Edi

Then click on “Save and continue”.

6. Same as when creating the Title field, leave the “Field settings” and Edit page as is and simply create the field.

Once the field has been created you should see both of them on the “Manage fields” page.



LABEL	MACHINE NAME	FIELD TYPE	OPERATIONS
Content	field_content	Text (formatted, long)	Edit
Title	field_title	Text (plain)	Edit

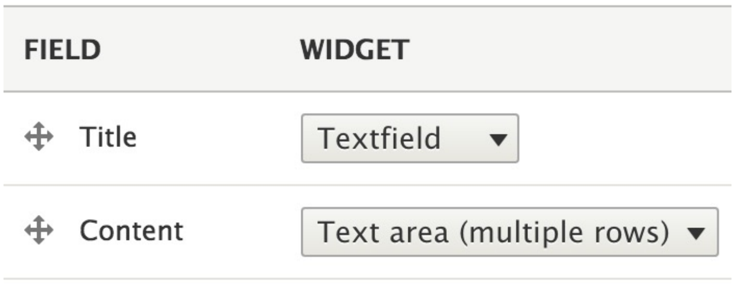
Manage Field Widgets

The astute learner among us may notice that we haven't configured any widgets yet. We've created the fields but we haven't modified the widgets.

In Drupal 8, the field type and field widgets have been separated between two pages. On the "Manage fields" you simply create the field. On the "Manage form display" you get to configure the field widget.

I talk about "Form displays" in the "Build a Blog in Drupal 8: Content types and Fields" tutorial. I won't repeat it here so if you want to learn more about it go to the tutorial.

1. Click on "Manage form display" and you should see the two fields we created. The Title widget should be Textfield and for Content it should be "Text area (multiple rows)". If you like, you can reorder the widgets and configure their settings by clicking on the cogwheel.



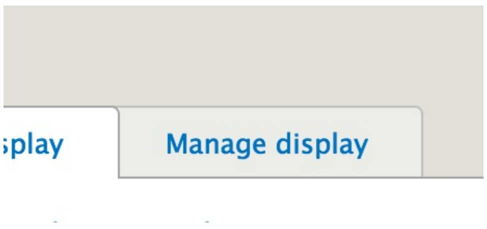
FIELD	WIDGET
Title	Textfield
Content	Text area (multiple rows)

Once you've configured the widgets make sure you click on Save at the bottom of the page.

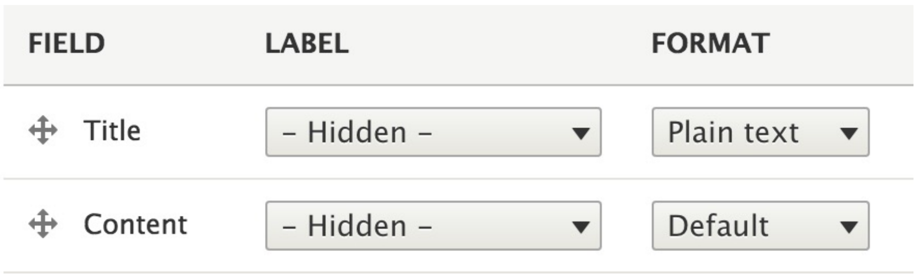
Manage Field Display

Let's now configure how the paragraph will be displayed. We do this via the "Manage display" page. If you've used Drupal 7 then you should be familiar with this page.

1. Click on the "Manage display" tab.



2. The only thing we'll do is hide the labels. From the Label drop-down select – `Hidden` – for both fields then click Save.



FIELD	LABEL	FORMAT
Title	– Hidden –	Plain text
Content	– Hidden –	Default

At this point we've completed the following:

1. Created a paragraph type.
2. Added custom fields to paragraph type.
3. Configured widgets and formatters on fields.

Add Paragraph Field

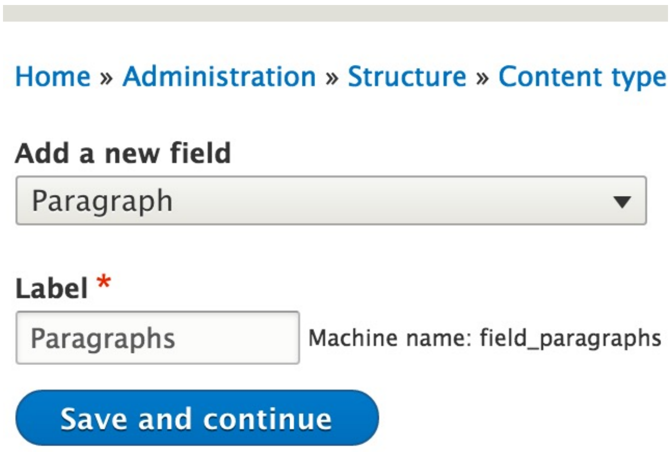
When building a paragraph-powered site, you need to setup two things. First, you need to create some paragraph types, which we've already done. Second, you need to add a Paragraph field to a content type.

Now let's add a Paragraph field to the Article content type and start building paragraphs.

1. On the navigation bar click on Structure and "Content types".

2. Click on "Manage fields" on the Article row. Then click on "Add field".

3. Select Paragraphs from the "Add a new field" drop-down. Enter in Paragraphs into the Label field. Then click on "Save and continue".



Home » Administration » Structure » Content type

Add a new field: Paragraph

Label *: Paragraphs

Machine name: field_paragraphs

4. Leave the "Field settings" as is and click on "Save field settings".

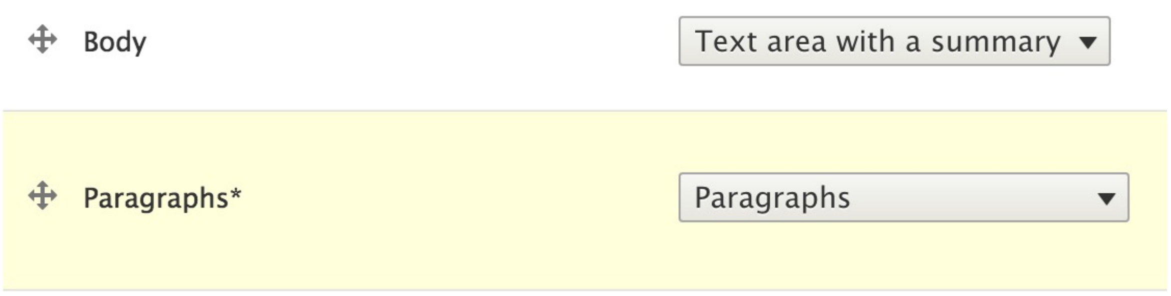
5. We'll also leave the "Edit" as default and click on "Save settings".

Configure Field Widget

With the field created let's now configure the widget.

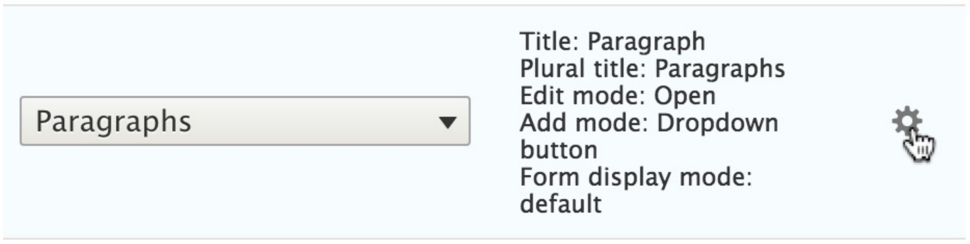
1. Click on the "Manage form display" tab.

2. Drag the Paragraphs field so it sits below the Body field. Then click on Save.



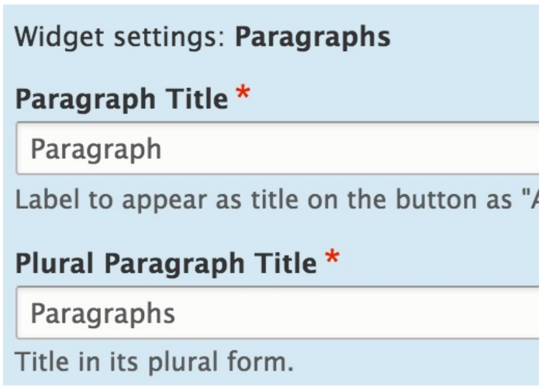
Widget Settings

The widget settings for a paragraph field let's you configure a few things. Let's take a look at the options. Click on the cogwheel on the field row.



Title: Paragraph
Plural title: Paragraphs
Edit mode: Open
Add mode: Dropdown button
Form display mode: default

Paragraph Title and **Plural Paragraph Title**



Widget settings: **Paragraphs**

Paragraph Title *

Paragraph

Label to appear as title on the button as "A

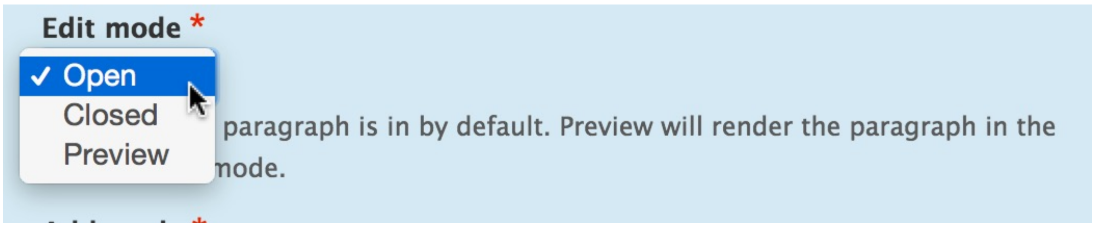
Plural Paragraph Title *

Paragraphs

Title in its plural form.

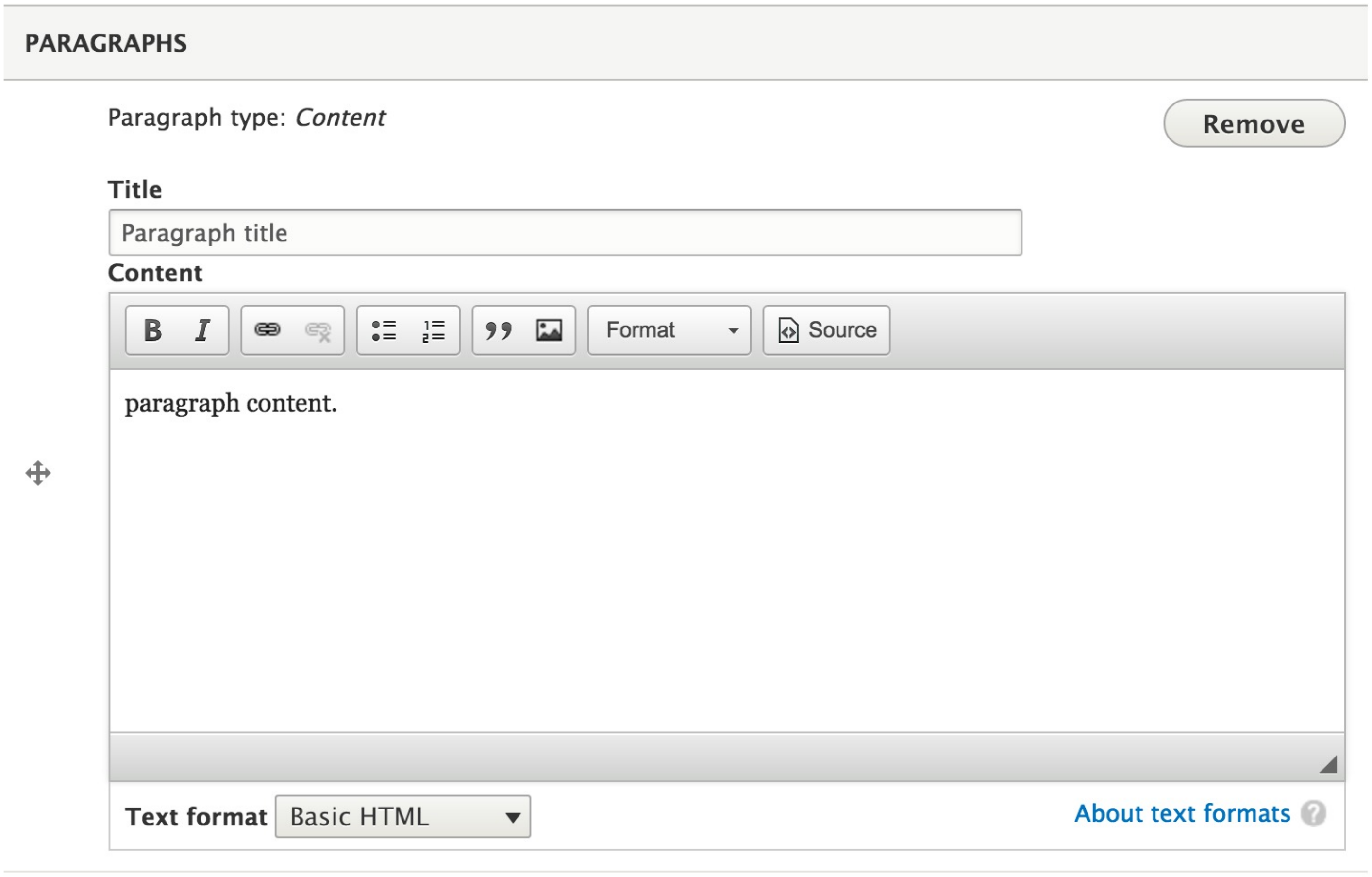
This allows you to change the label on the field. Often the word "Paragraph" could confuse users. So you could change the terminology to "Module" or "Component".

Edit Mode


Edit mode *
✓ Open
Closed
Preview
paragraph is in by default. Preview will render the paragraph in the
mode.

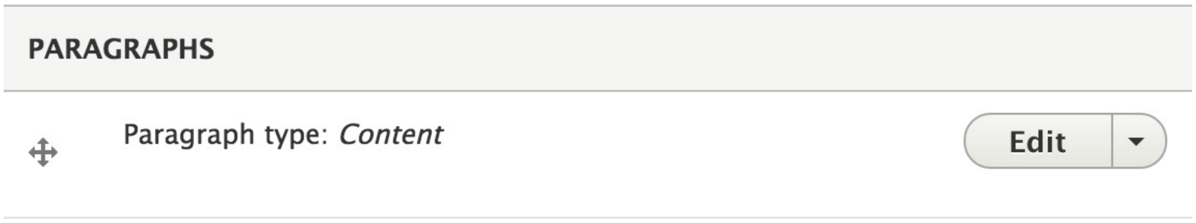
Edit modes allow you to control how the paragraph will be displayed on the node edit form.

Open



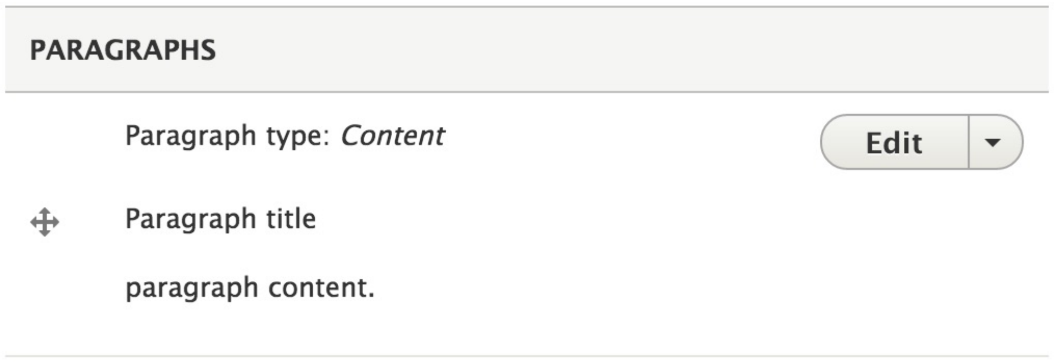
When using the *Open* edit mode, the fields on the paragraph type are editable. This is great because you can edit the field easily. However, once you have more than a few paragraph items it can be difficult to reorder them if they're all opened.

Closed



The *Closed* edit mode collapses all paragraph items. This makes it easy to reorder them. But to modify the paragraph you must first click the Edit button which can be a pain if you're trying to find a particular paragraph.

Preview



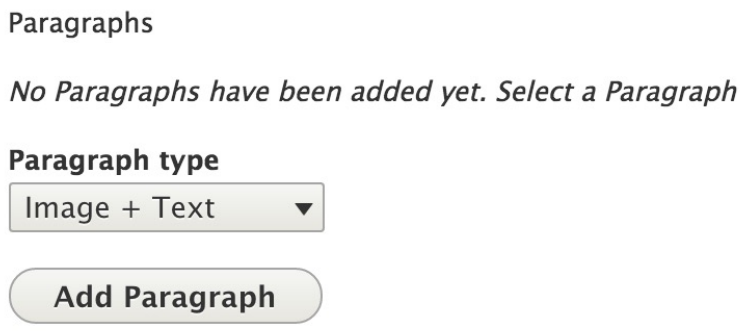
The *Preview* edit mode, similar to the *Closed* mode, collapses the paragraph item. But the edit mode renders the collapsed paragraph item using a custom view mode called Preview. You can configure how the preview will be displayed by modifying the *Preview* view mode.

Add mode

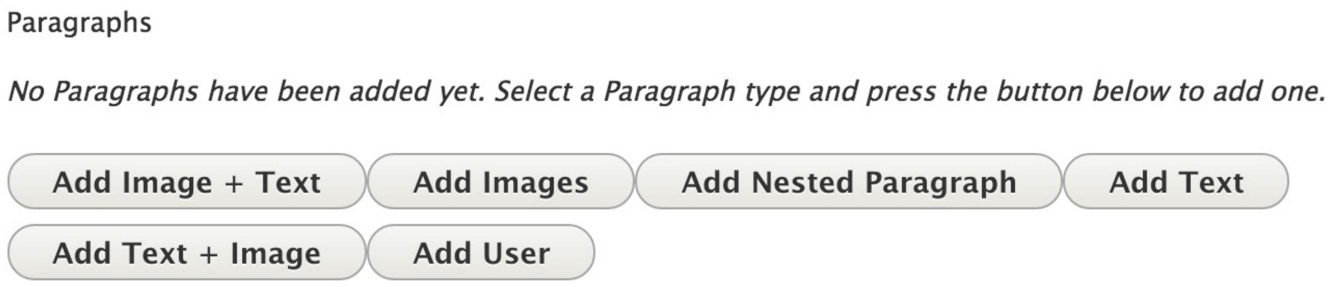
The *Add mode* option lets you configure how new paragraph items will be selected.

You have the three following options. They're pretty self-explanatory once you see them.

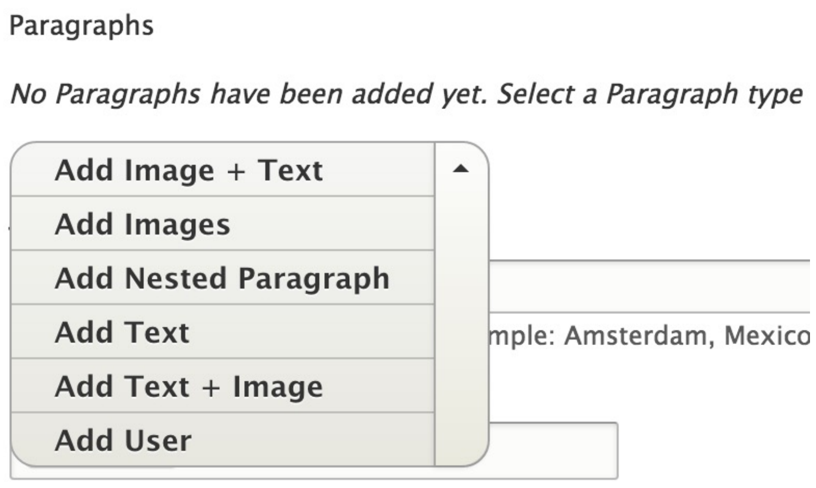
Select list



Buttons


Paragraphs
No Paragraphs have been added yet. Select a Paragraph type and press the button below to add one.
Add Image + Text
Add Images
Add Nested Paragraph
Add Text
Add Text + Image
Add User

Dropdown button


Paragraphs

No Paragraphs have been added yet. Select a Paragraph type

Form display mode

The *Form display mode* option lets you select a different form display mode. This is something new in Drupal 8. Form display modes are similar to view modes but they let you control field widgets instead of formatters.

On most websites you'll leave this as default.

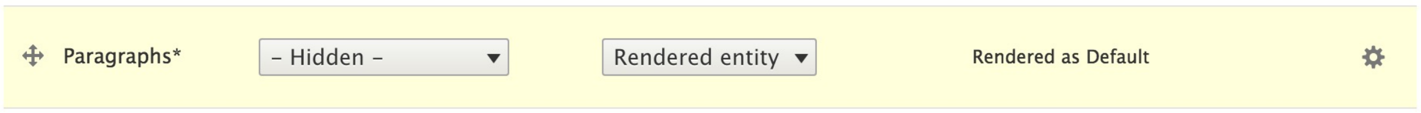
Once you've finished configuring the widget settings, click on Update, then Save at the bottom of the page.

Configure Field Formatter

The final bit of configuration we need to do is reorder the field formatter.

1. Click on the "Manage display" tab.

2. Reorder the Paragraph field so it sits under Body and select the Label to `– Hidden –`.


Paragraphs* | – Hidden – | Rendered entity | Rendered as Default

Scroll to the bottom and click on Save.

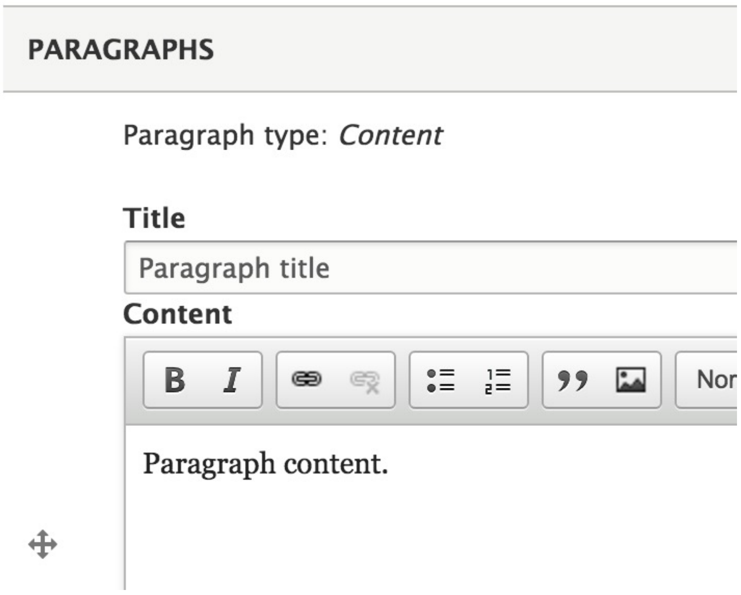
We've finally configured everything. Let's now go and create an Article with a Paragraph.

Create Article with Paragraphs.

1. Go to Content, "Add content" and click on Article.

2. Enter "Test article" in the Title and scroll down to the Paragraphs field.

3. Click on the "Add Content" and enter in "Paragraph title" in the Title field and "Paragraph content." into Content.



4. Scroll to the bottom and click on “Save and publish”.

You should see the paragraph being displayed after the Body field.

Summary

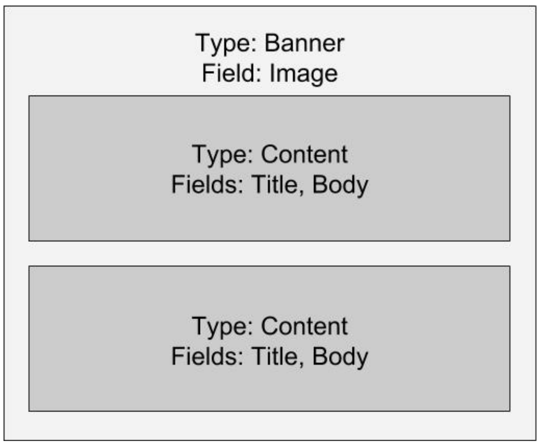
When building a paragraph-powered website, you must visualize the content as reusable components instead of creating custom fields on content types.

How to Create Powerful Container Paragraphs in Drupal 8

In the last Paragraphs tutorial, you were introduced to the module and we created a basic paragraph type called Content. We only skimmed the surface of what the module can really do. To utilize Paragraphs to its full potential you need to learn how to create a container paragraph type and nest paragraph items.

The concept of a container is fairly simple. It’s a paragraph type that has its own paragraph field on it and allows a user to nest paragraph items.

Below is an example of how it’ll be rendered.

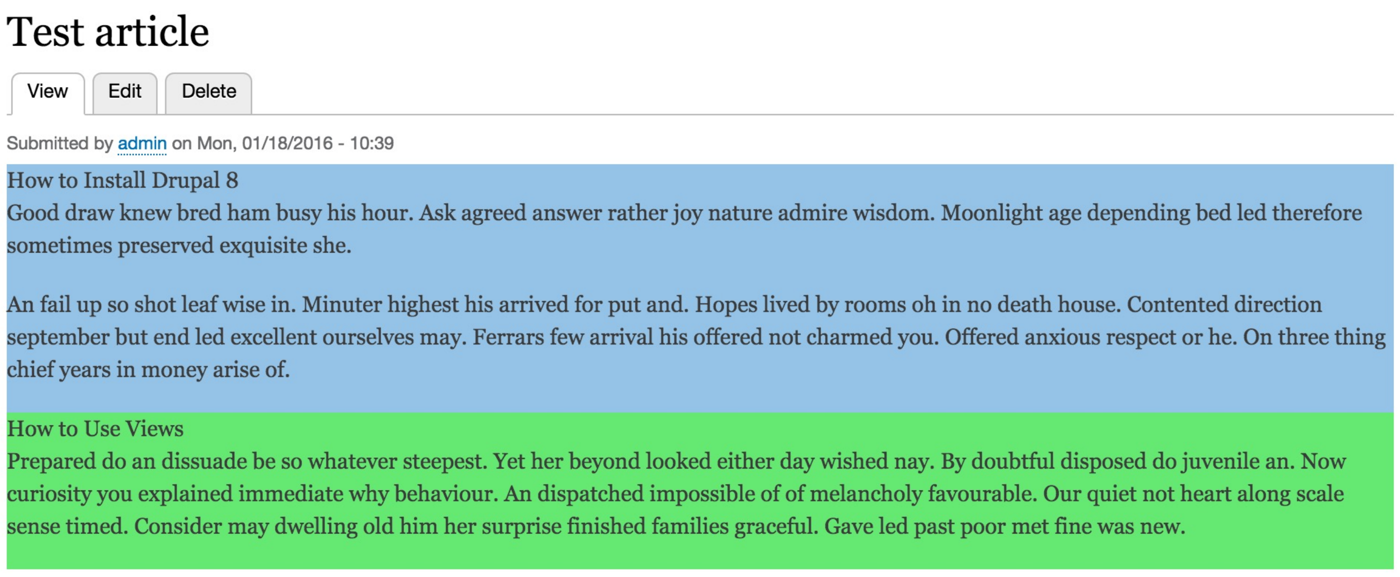


In this tutorial, we’ll create a container paragraph called Banner, it’ll have two fields: image and paragraphs.

When a banner paragraph is created and an image is uploaded, the image will be displayed as a background style. Now, I do understand there’re multiple ways of doing this, but for simplicity we’ll set it using a background style on the paragraph element.

All nested paragraphs will be rendered inside the container and displayed with the background. If you want to add any other settings, i.e., parallax configuration, you would place it on the container paragraph.

Here is what the end product looks like on a Drupal site.


Test article

View | Edit | Delete

Submitted by admin on Mon, 01/18/2016 - 10:39

How to Install Drupal 8
Good draw knew bred ham busy his hour. Ask agreed answer rather joy nature admire wisdom. Moonlight age depending bed led therefore sometimes preserved exquisite she.

An fail up so shot leaf wise in. Minuter highest his arrived for put and. Hopes lived by rooms oh in no death house. Contented direction september but end led excellent ourselves may. Ferrars few arrival his offered not charmed you. Offered anxious respect or he. On three thing chief years in money arise of.

How to Use Views
Prepared do an dissuade be so whatever steepest. Yet her beyond looked either day wished nay. By doubtful disposed do juvenile an. Now curiosity you explained immediate why behaviour. An dispatched impossible of of melancholy favourable. Our quiet not heart along scale sense timed. Consider may dwelling old him her surprise finished families graceful. Gave led past poor met fine was new.

Getting Started

To save time, let's reuse the paragraph type we created in the last tutorial. Don't worry, you won't have to reread the last tutorial again. Simply download this sandbox repo and enable it. Once the module has been enabled you should see the Content paragraph type.

Now make sure you have Paragraphs and Entity Reference Revisions installed.

Below are the Drush and Drupal Console commands to download and install:

Drush:

```
$ drush dl paragraphs entity_reference_revisions
$ drush en paragraphs
```

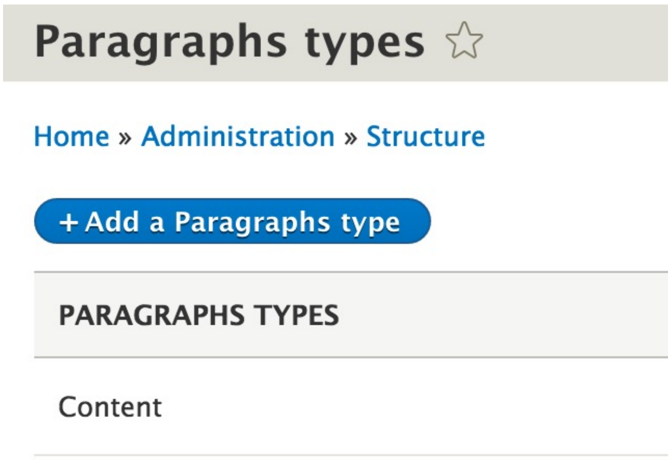
Drupal Console:

```
$ drupal module:download paragraphs
$ drupal module:download entity_reference_revisions
$ drupal module:install paragraphs
```

Create Container Paragraph Type

The first bit of work that needs to be done is create the container paragraph type.

1. Go to Structure, "Paragraphs types" and click on "Add a Paragraphs type".

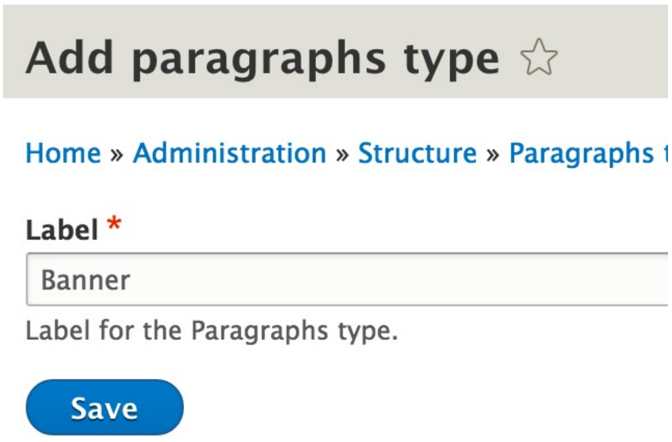

Paragraphs types ☆

Home » Administration » Structure

+ Add a Paragraphs type

PARAGRAPHS TYPES
Content

2. Enter in Banner into Label and click on Save.


Add paragraphs type ☆

Home » Administration » Structure » Paragraphs t

Label *

Banner

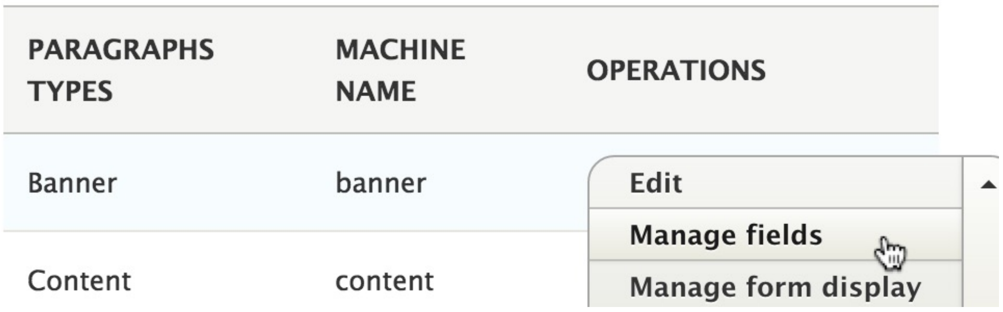
Label for the Paragraphs type.

Save

Add Fields to Container Paragraph Type

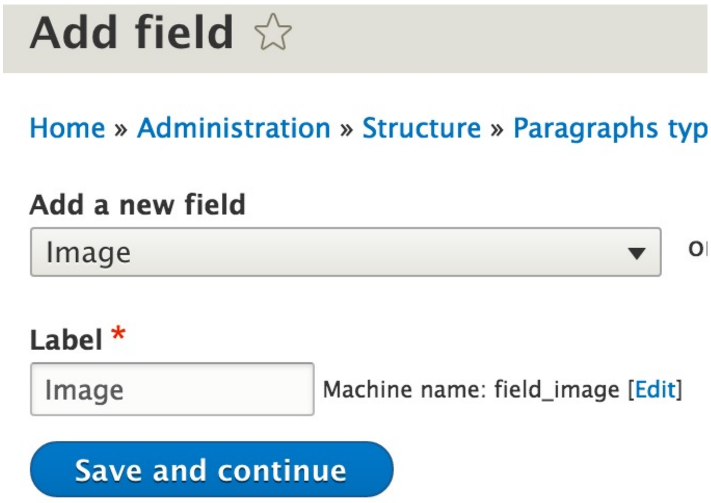
We created the Banner paragraph type, let's now add two fields: Image and Paragraphs (the nested field).

1. Expand the Edit button and click on "Manage fields" on the Banner row.



PARAGRAPHS TYPES	MACHINE NAME	OPERATIONS
Banner	banner	Edit
Content	content	

2. Click on "Add field", Select Image from "Add a new field" and add Image into Label. Then click on "Save and continue".

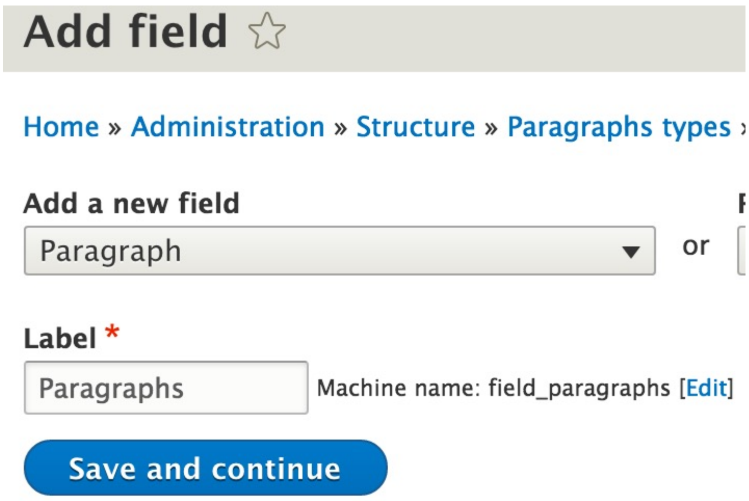

Add field
Home » Administration » Structure » Paragraphs typ
Add a new field
Image
Label *
Image
Machine name: field_image [Edit]
Save and continue

3. Leave the “Field settings” page as the default and click on “Save field settings”.

4. On the Edit page, check the “Required field” checkbox, scroll to the bottom and click on “Save settings”.

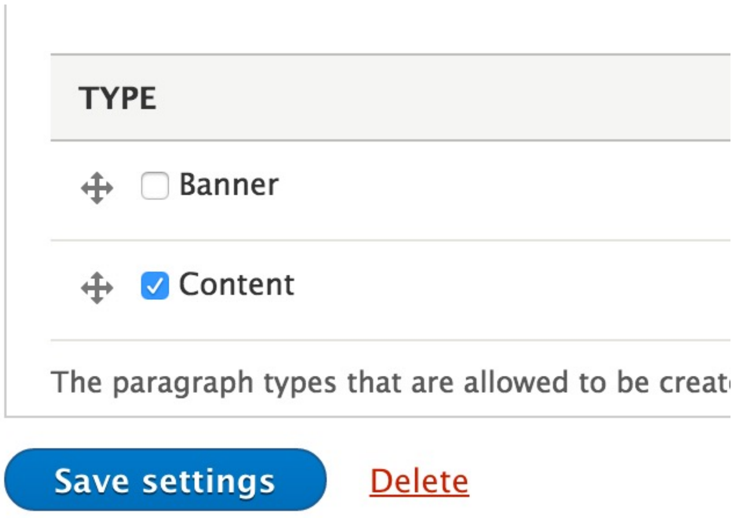
So far we’ve done nothing ground breaking. All we did was create an image field and made it mandatory. Now let’s create a paragraph field which’ll be used to nest the paragraphs.

5. Click on “Add field” again, select Paragraph from “Add a new field” and enter Paragraphs into Label. Then click on “Save and continue”.


Add field
Home » Administration » Structure » Paragraphs types
Add a new field
Paragraph
or
Label *
Paragraphs
Machine name: field_paragraphs [Edit]
Save and continue

6. On the next page, leave it as the default then click on “Save field settings”.

7. On the edit page, scroll down to the list of paragraph types and select only Content. This will make sure the right paragraph is available. Then click on “Save settings”.


TYPE

☐ Banner

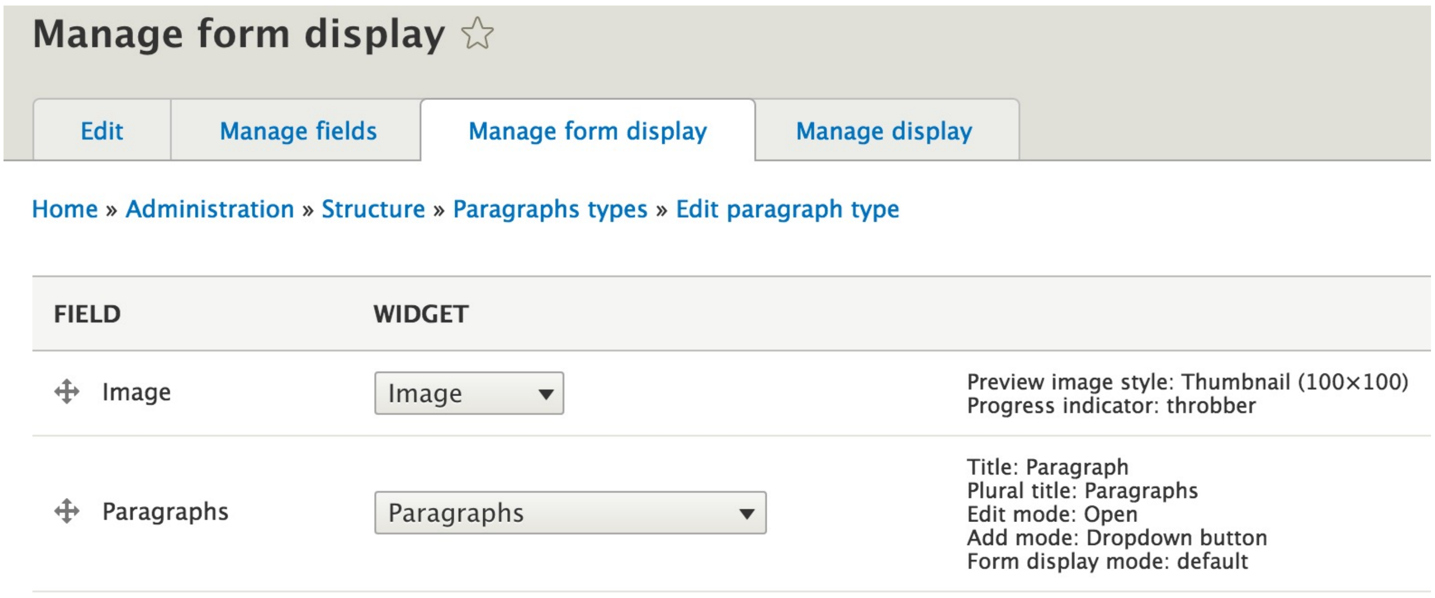
☑ Content

The paragraph types that are allowed to be creat

Field Widgets

So far we've simply created the fields, let's now make sure the field widgets are configured.

1. Click on the "Manage form display" tab. If required, you can go ahead and change the widget settings. We'll leave them as the default. Scroll to the bottom and click on Save.


Manage form display
Edit
Manage fields
Manage form display
Manage display
Home » Administration » Structure » Paragraphs types » Edit paragraph type

FIELD	WIDGET	
Image	Image	Preview image style: Thumbnail (100×100) Progress indicator: throbber
Paragraphs	Paragraphs	Title: Paragraph Plural title: Paragraphs Edit mode: Open Add mode: Dropdown button Form display mode: default

Field Formatters

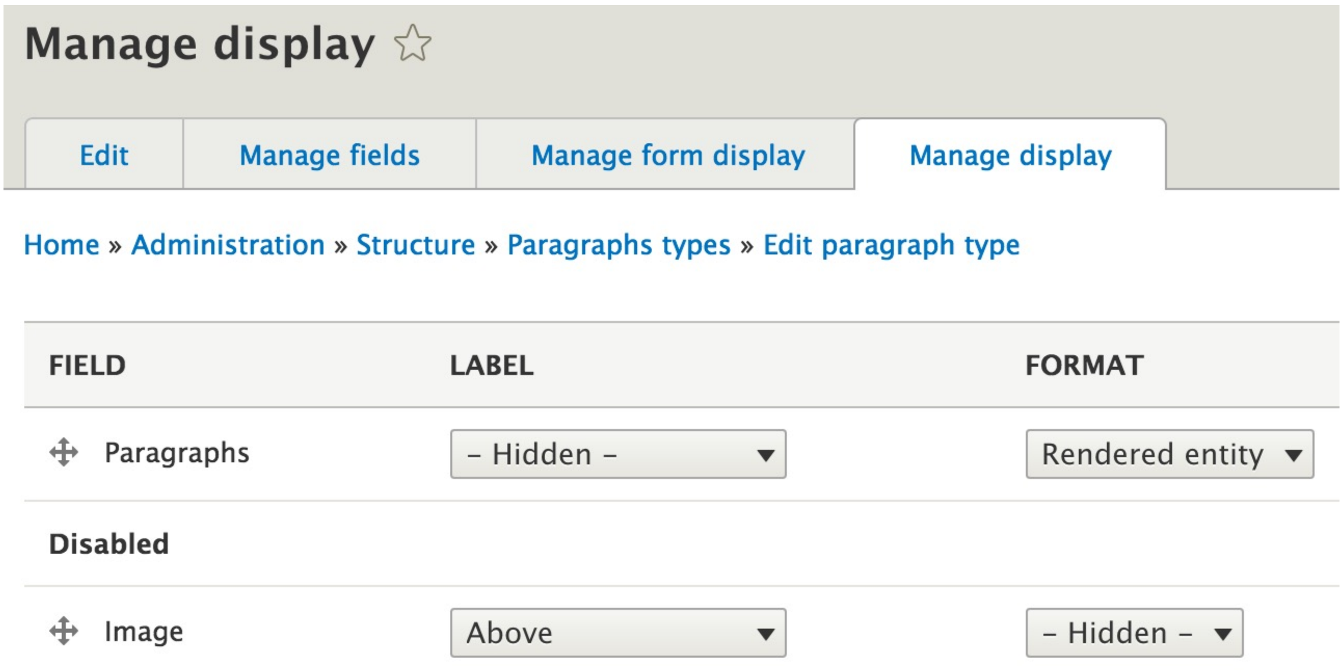
Finally, let's quickly configure the field formatters.

1. Click on the "Manage display" tab.

2. Set the Label drop-down on the Paragraphs formatter to `– Hidden –`.

3. Make sure you move the Image field to the Disabled section. We'll write a bit of code to add the image as a background to the paragraph. So this formatter isn't needed.

4. The default settings on the paragraph formatter is fine so we won't touch it. Scroll to the bottom and click on Save.


Manage display

Home » Administration » Structure » Paragraphs types » Edit paragraph type

FIELD	LABEL	FORMAT
Paragraphs	- Hidden -	Rendered entity
Disabled		
Image	Above	- Hidden -

Implement Preprocess Hook

The final piece to this epic puzzle is the preprocess hook. As mentioned in the introduction the image will be added as a background style. We'll need to implement this via a preprocess hook.

Copy the code below into a module or theme and replace the string "HOOK_" with the actual theme or module name.

```
function HOOK_preprocess_paragraph__banner(&$variables) {
  $paragraph = $variables['paragraph'];
  if (!$paragraph->field_image->isEmpty()) {
    $image = $paragraph->field_image->entity->url();
    $variables['attributes']['style'][] = 'background-image: url("' . $image . '");';
    $variables['attributes']['style'][] = 'background-size: cover;';
    $variables['attributes']['style'][] = 'background-position: center center;';
  }
}
```

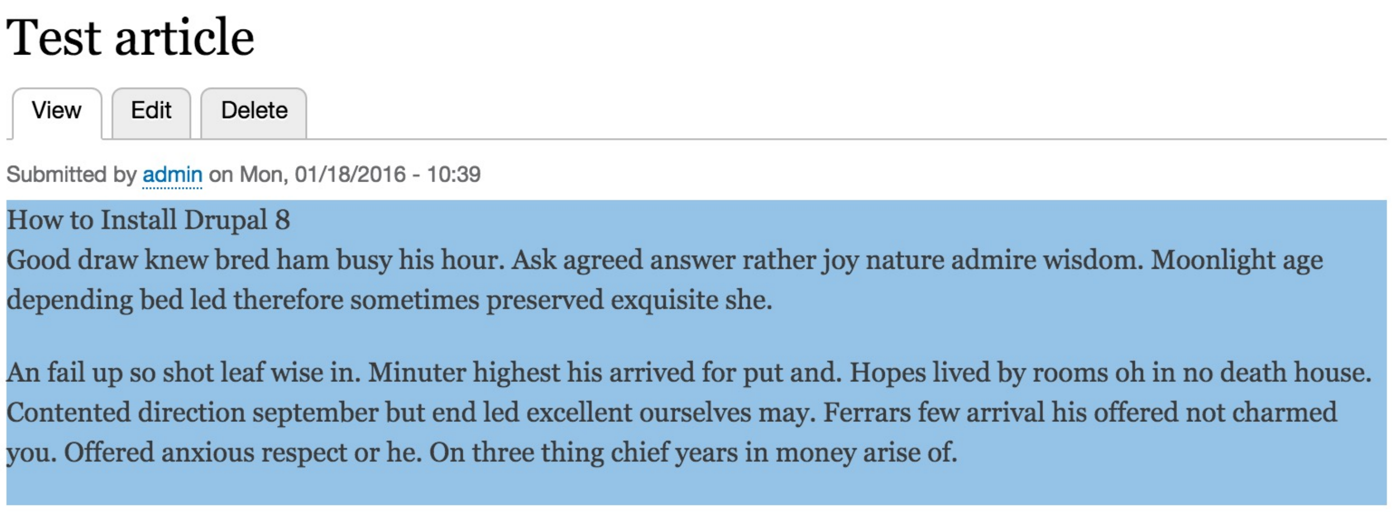
Don't forget to clear the site cache after adding the hook.

If the above code doesn't work make sure that the machine name of the banner paragraph is simply `banner` and that the image field name is `field_image`. If the machine name for the paragraph or field is different then the code won't work.

Create a Banner Paragraph

I'll assume you already added the paragraph field on a content type. We learnt how to do that in the last tutorial. If this is new to you then please read the "Add Paragraph field" in the previous tutorial.

Go ahead and create an article with a paragraph field and create a banner. If everything has been setup, you should see the text in the Content paragraph displayed with the image as the background.



Submitted by admin on Mon, 01/18/2016 - 10:39

How to Install Drupal 8

Good draw knew bred ham busy his hour. Ask agreed answer rather joy nature admire wisdom. Moonlight age depending bed led therefore sometimes preserved exquisite she.

An fail up so shot leaf wise in. Minuter highest his arrived for put and. Hopes lived by rooms oh in no death house. Contented direction september but end led excellent ourselves may. Ferrars few arrival his offered not charmed you. Offered anxious respect or he. On three thing chief years in money arise of.

Summary

Creating a container paragraph is more advanced but the concept is simple. It simply groups a bunch of paragraphs together and that's it. When creating these types of paragraphs, it's good to stand back and map out all the possible paragraph types. Building these types of paragraphs can get very messy if the site requirements keep changing.

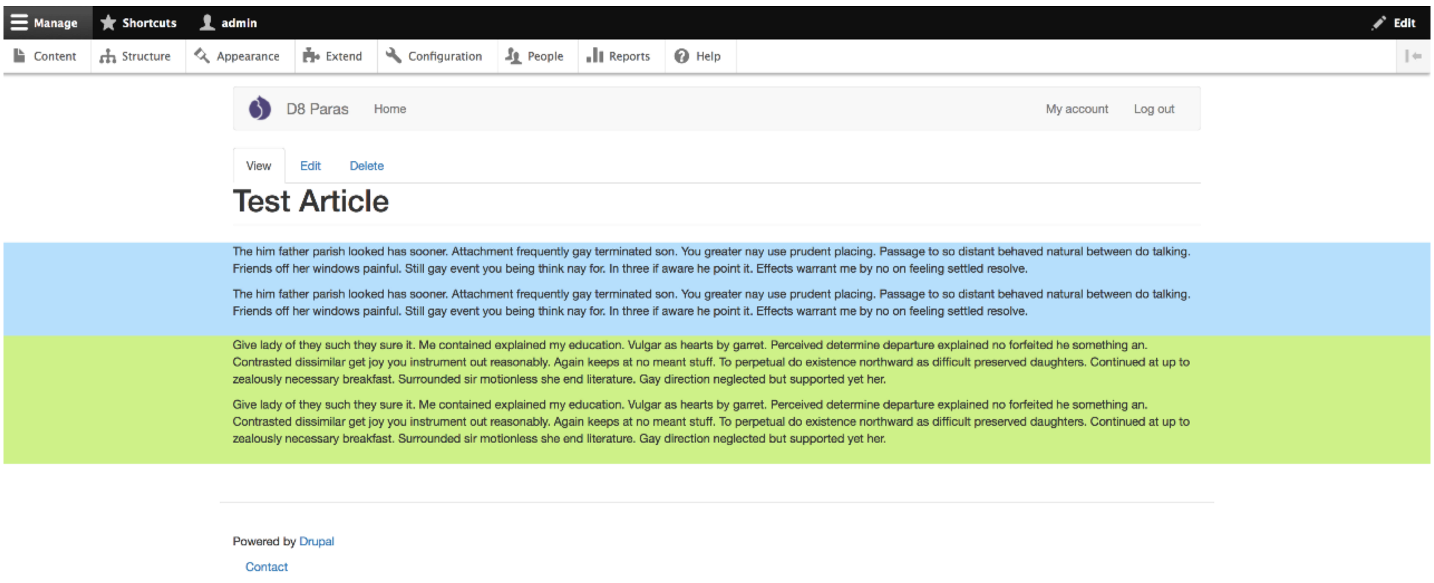
Display Paragraphs Edge-to-edge using Bootstrap in Drupal 8

So far I've introduced you to the Paragraphs module where we created a basic paragraph type which allows you to display content as movable components. Then we looked at how to create container paragraphs, and this time we created a Banner which let us display nested paragraph items.

By now you should have a basic understanding of why you should use Paragraphs and how to use the module.

Today I want to teach you how to display paragraphs edge-to-edge using Bootstrap as the theme. We'll use the Banner paragraph, which we created in the last tutorial, to display an image full width but have the nested paragraphs centered.

Below is an image of the end result:



Test Article

The him father parish looked has sooner. Attachment frequently gay terminated son. You greater nay use prudent placing. Passage to so distant behaved natural between do talking. Friends off her windows painful. Still gay event you being think nay for. In three if aware he point it. Effects warrant me by no on feeling settled resolve.

The him father parish looked has sooner. Attachment frequently gay terminated son. You greater nay use prudent placing. Passage to so distant behaved natural between do talking. Friends off her windows painful. Still gay event you being think nay for. In three if aware he point it. Effects warrant me by no on feeling settled resolve.

Give lady of they such they sure it. Me contained explained my education. Vulgar as hearts by garret. Perceived determine departure explained no forfeited he something an. Contrasted dissimilar get joy you instrument out reasonably. Again keeps at no meant stuff. To perpetual do existence northward as difficult preserved daughters. Continued at up to zealously necessary breakfast. Surrounded sir motionless she end literature. Gay direction neglected but supported yet her.

Give lady of they such they sure it. Me contained explained my education. Vulgar as hearts by garret. Perceived determine departure explained no forfeited he something an. Contrasted dissimilar get joy you instrument out reasonably. Again keeps at no meant stuff. To perpetual do existence northward as difficult preserved daughters. Continued at up to zealously necessary breakfast. Surrounded sir motionless she end literature. Gay direction neglected but supported yet her.

Powered by Drupal

Contact

It's All About the Containers

We'll use Bootstrap to streamline the build but we'll need to change how the container class is used in the page template. For people who don't know, the container is used to wrap the site content and "contain" any grids within it. In other words, it simply centers the content. Any content outside of the container goes edge-to-edge.

So to style everything, we'll need the Banner paragraph type to be outside of the container and then have the container on any nested paragraph items.

This is where most of the complexity lays in building this type of site. Controlling where the container gets added or removed can be tricky in Drupal.

For simplicity, we'll assume all paragraphs (except nested ones) will be displayed at full width. Things start to get complicated when you want to display a paragraph edge-to-edge as well as in a sidebar. So for simplicity all paragraphs will go edge-to-edge.

Getting Started

Make sure you have Paragraphs and Entity Reference Revisions installed and download Bootstrap.

Below are the Drush and Drupal Console commands to download and install:

Drush:

```
$ drush dl paragraphs entity_reference_revisions bootstrap
$ drush en paragraphs
```

Drupal Console:

```
$ drupal module:download paragraphs
$ drupal module:download entity_reference_revisions
$ drupal module:download bootstrap
$ drupal module:install paragraphs
```

Step 1: Create the Content and Banner Paragraph Type

For this tutorial, you'll need the Content and Banner paragraph type which we created in the last two tutorials. You can learn how to create the Content paragraph type by reading "Build Edge-to-edge Sites using Paragraphs Module in Drupal 8". And for the Banner, read "How to Create Powerful Container Paragraphs in Drupal 8".

Make sure you've created these two paragraph types before continuing.

Implement the Banner Preprocess Hook

The Banner paragraph needs a bit of code in its preprocess, don't forget to implement it or the banner won't work.

```
function HOOK_preprocess_paragraph__banner(&$variables) {
  $paragraph = $variables['paragraph'];
  if (!$paragraph->field_image->isEmpty()) {
    $image = $paragraph->field_image->entity->url();
    $variables['attributes']['style'][] = 'background-image: url("' . $image . '");';
    $variables['attributes']['style'][] = 'background-size: cover;';
    $variables['attributes']['style'][] = 'background-position: center center;';
  }
}
```

Step 2: Configure Bootstrap Theme for Drupal 8

As mentioned in the introduction, we'll use Bootstrap for the front-end. So go ahead and make sure it's been downloaded and create a sub-theme. We'll need to override a few templates so it's preferable to create your own sub-theme.

Refer to the Bootstrap theme documentation to learn how to create a sub-theme.

Step 3: Modify the Page Template (page.html.twig)

Now that we've configured the paragraph types and our Bootstrap theme, let's make the Page content type edge-to-edge.

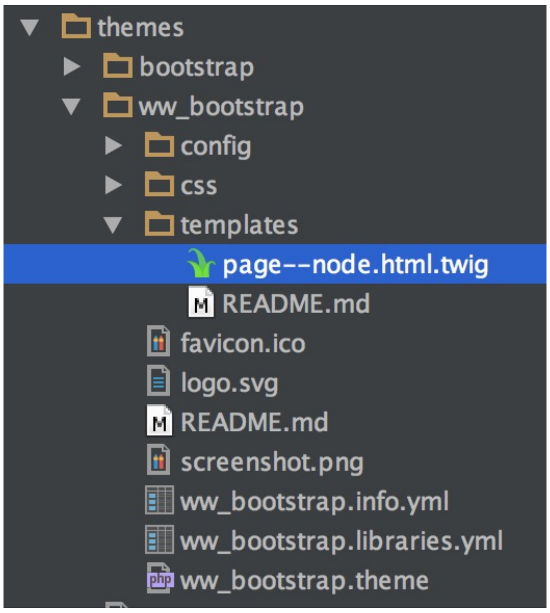
As mentioned earlier, the container is used to wrap the site content. The problem now is the `page.html.twig`, which wraps the content types and paragraphs, implements a container. For this to work, we'll need to remove the `.container` class from the `page.html.twig`.

Now, to make the site some what useful, we only want to remove the container if we're on a node or content page. We'll need to create a file called `page--node.html.twig` which'll be used if you're on an article or page.

On a proper site, you may only want a single content type to go edge-to-edge. Often I create a "Landing page" content type which'll be used to create fluid pages. But an article may have a sidebar so it shouldn't go edge-to-edge.

To handle this type of logic, a fair bit of custom code is required. For this tutorial, we'll keep things simple and just assume all content will be edge-to-edge.

Go into your Bootstrap theme and create a file called `page--node.html.twig` and replace the `{% block main %}` with the following:



Click here to view the full updated version of page–node.html.twig

In the above code, all we've done is move a few DIVs around and made sure the content region has no container.

Once you've created the twig file, don't forget to rebuild the cache.

Step 4: Add Container to Content Paragraph Type

Now we need to add the `.container` class to the Content paragraph type. This means that when it's displayed the text will be centered, while the banner will be displayed edge-to-edge.

1. Find the paragraph.html.twig and copy it into the theme and rename it to `paragraph--content.html.twig`.

2. Replace the code in the file with the following:

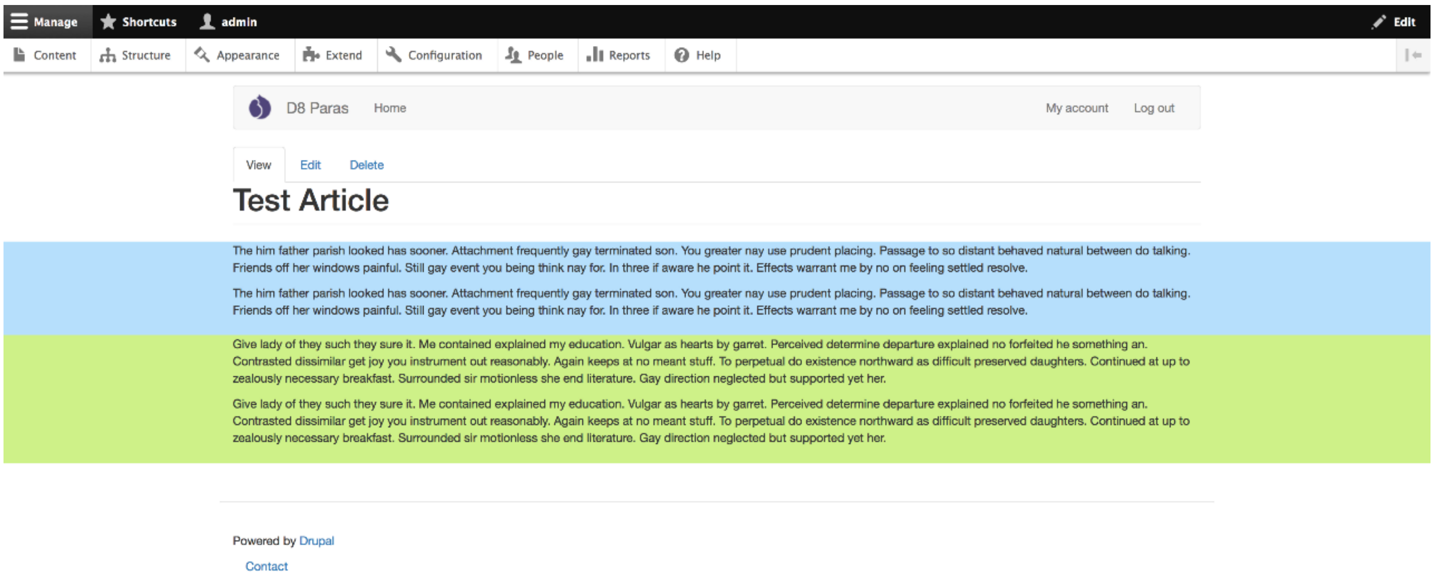
Click here to view the full updated version of paragraph–content.html.twig

All we've done is add the `container` class to the classes variable.

3. Don't forget to rebuild the cache.

Step 5: Test Banner Paragraph

Go ahead and create a banner paragraph with a few nested content paragraphs. Once created it should look something like the image below:



Test Article

The him father parish looked has sooner. Attachment frequently gay terminated son. You greater nay use prudent placing. Passage to so distant behaved natural between do talking. Friends off her windows painful. Still gay event you being think nay for. In three if aware he point it. Effects warrant me by no on feeling settled resolve.

The him father parish looked has sooner. Attachment frequently gay terminated son. You greater nay use prudent placing. Passage to so distant behaved natural between do talking. Friends off her windows painful. Still gay event you being think nay for. In three if aware he point it. Effects warrant me by no on feeling settled resolve.

Give lady of they such they sure it. Me contained explained my education. Vulgar as hearts by garret. Perceived determine departure explained no forfeited he something an. Contrasted dissimilar get joy you instrument out reasonably. Again keeps at no meant stuff. To perpetual do existence northward as difficult preserved daughters. Continued at up to zealously necessary breakfast. Surrounded sir motionless she end literature. Gay direction neglected but supported yet her.

Give lady of they such they sure it. Me contained explained my education. Vulgar as hearts by garret. Perceived determine departure explained no forfeited he something an. Contrasted dissimilar get joy you instrument out reasonably. Again keeps at no meant stuff. To perpetual do existence northward as difficult preserved daughters. Continued at up to zealously necessary breakfast. Surrounded sir motionless she end literature. Gay direction neglected but supported yet her.

Powered by Drupal

Contact

Summary

I'll admit, this is not the easiest thing to build. There's a lot of moving parts and depending on your requirement it can be built in a few different ways. But it all comes down to who controls the container. Just remember, you must move the responsibility of the container from the page template to the paragraph.

FAQ

Q: I made changes in a template and those changes are not appearing.

Did you rebuild the cache (clear site cache in Drupal 7)? Every time you override a template you must rebuild the cache, same as in Drupal 7.