anagramballoons.com Case Study

MINNESOTA KENTICO USER GROUP – 11/15/2017

DAHLIN DEVELOPMENT

Dahlin Development

- Established in 2008
- Kentico Developers, 3 certified
- Over 75 sites built on Kentico
- Partner with other Kentico partners

Anagram Balloons

Anagram leads the world in making balloons fun with consumer-inspired - and inspiring - product development, industry-expanding innovation and strong, value-added partnerships.

We proudly offer many products you'll only find with an Anagram label. Our dedicated-to-the joy employees are focused on creating the best product and the biggest smiles possible.

At Anagram, we strive to create joy, on purpose, every single day.

Brand Video

Goals

- Build the team

Technical

- Environment Setup
- Digital Asset Management (DAM) integration
- Modular Design
  - Widgets
  - Page Types

Team Building

- Parties
  - Curb Crowser (agency)
  - Anagram Balloons (client)
  - Hello Wallace (design/frontend)
  - Dahlin Development (Kentico development)
- Set Expectations
- Agree upon tools
- Use what works

Our Collaboration Toolset

- Skype
- OneNote (internal)
- OneDrive (internal)
- Bitbucket
- invision
- Trello
- Google Sheets
- UberConference
- Slack
- E-mail
- Phone

Communication is Key

- Initial in-person meeting
- Project done all remotely
- Weekly Standups
- Ad-hoc meetings
- Near the end daily standups
- Slack can be nice

Environment Setup

- Anagram development server was behind a VPN.
- Because of early access issues, we decided to develop outside the VPN and sync content over to the development server.
- Spun up a database in Azure and imported the site from the development server into the azure database.
- All development done locally connecting to Azure database.
- When changes are ready, they were synced to the development server using the Kentico staging module.

Digital Asset Management (DAM) Integration

- Previously, all products on the site were manually maintained.
- On the new site, the following pages display data from the DAM:
- https://anagramballoons.com/
- https://anagramballoons.com/products/
- https://anagramballoons.com/products/airwalkers/ - multiple pages per category
- https://anagramballoons.com/search/?q=batman

- The DAM returns JSON, which is deserialized into class objects using the Json.Net framework for .Net (https://www.newtonsoft.com/json)
- All DAM calls use the same model classes and are cached using Kentico caching.

Digital Asset Management (DAM) Integration

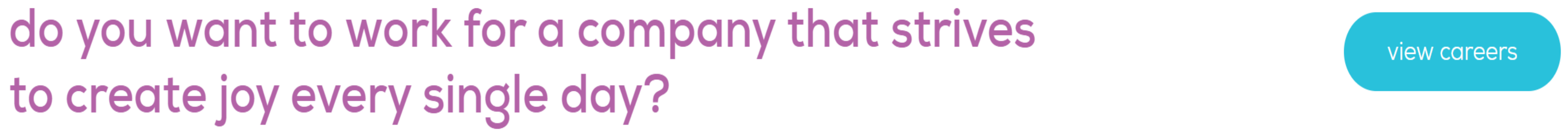
- The product page templates are built using three custom web parts – featured products, products, and search results.
- All three web parts use the same common user control to display the results.
- Each web part retrieves the product data it needs and then sets the data source on the control.

```
ProductsList.assets = Asset.GetAssetsByCategoryKey(CurrentDocument.GetStringValue("AdamKey", string.Empty));
```

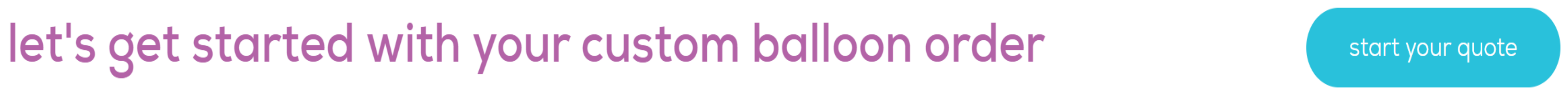
- The common user control uses a basic repeater to display the results and a unipager for paging.

Modular Design – Widgets

- Going through the site, you will notice that many of the pages share the same visual components
- About page:

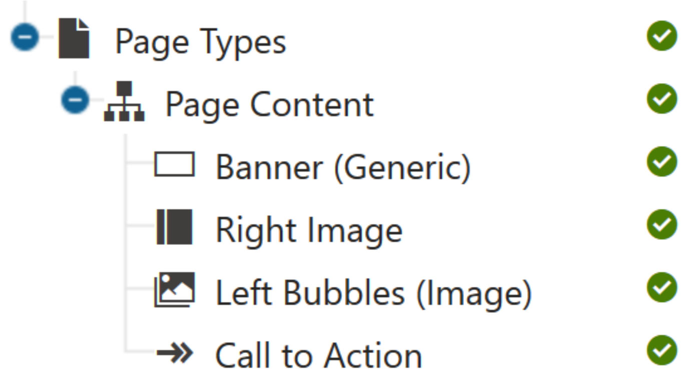

do you want to work for a company that strives to create joy every single day?

- Custom Balloons:


let's get started with your custom balloon order

Modular Designs - Widgets

On previous projects we achieved modular design by using page types and a hierarchical repeater in the content tree.



For Anagram, Rui Wang (Kentico Consulting) suggested a different approach that uses widgets.

Rui's Demo: https://www.youtube.com/watch?v=voumgLD-R1Y

We converted all of the modular page types into widgets to use the new approach.

Modular Designs - Widgets

Each Widget has two parts.

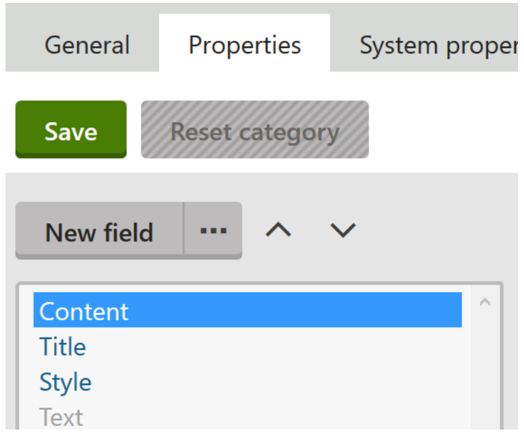
- Widget – input fields
- Container - markup

The widget and container work together to render the html output.

Modular Designs - Widgets

Widget

- Create Widgets in the Widget application of Kentico
- The easiest way to start is to clone a simple Kentico widget, we used **Static text**.
- Once the base widget is cloned, add custom properties to the new widget

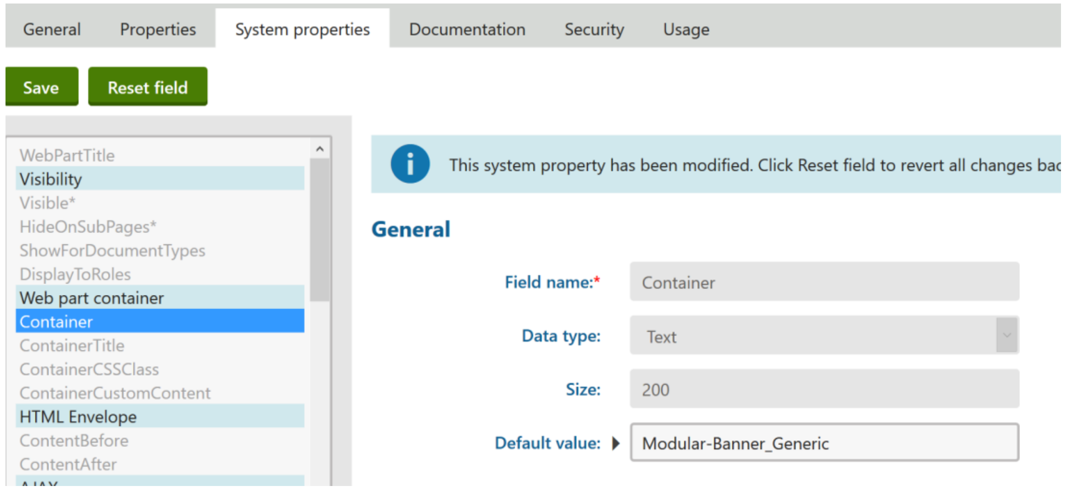


Content
Title
Style
Text

Modular Designs - Widgets

Widget

- On the System Properties tab, default the Container to the web part container that will render the widget. This will default the widget to the correct container when adding to a page.
- Give the container the same name as the widget.



This system property has been modified. Click Reset field to revert all changes bac

General

Field name:* Container

Data type: Text

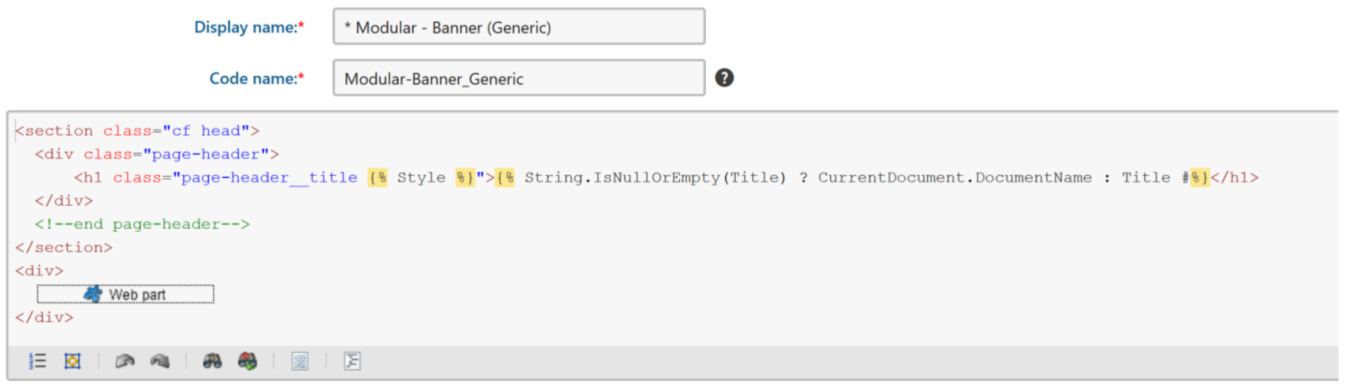
Size: 200

Default value: Modular-Banner_Generic

Modular Designs - Widgets

Container

- Create Containers in the Web part containers application of Kentico
- Give the container the same name as the widget
- Enter the markup that should be rendered for the widget. Macros can be used.



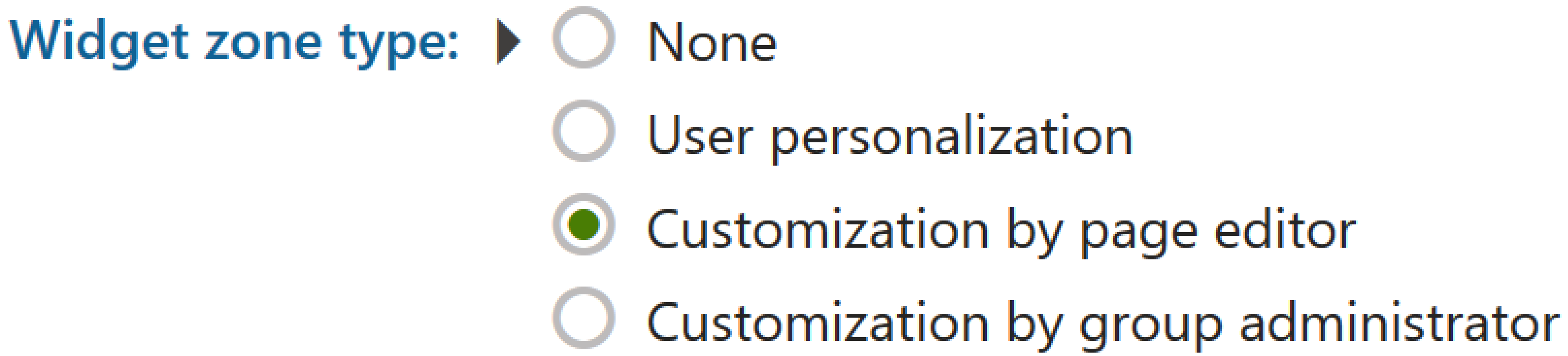
Display name:*	* Modular - Banner (Generic)
Code name:*	Modular-Banner_Generic

```
<section class="cf head">
  <div class="page-header">
      <h1 class="page-header__title {% Style %}">{% String.IsNullOrEmpty(Title) ? CurrentDocument.DocumentName : Title #%}</h1>
  </div>
  <!--end page-header-->
</section>
<div>
   Web part
</div>
```

Modular Designs - Widgets

Page Template

- On the page template that will display widgets, add a web part zone
- Set the widget zone type to **Customization by page editor**



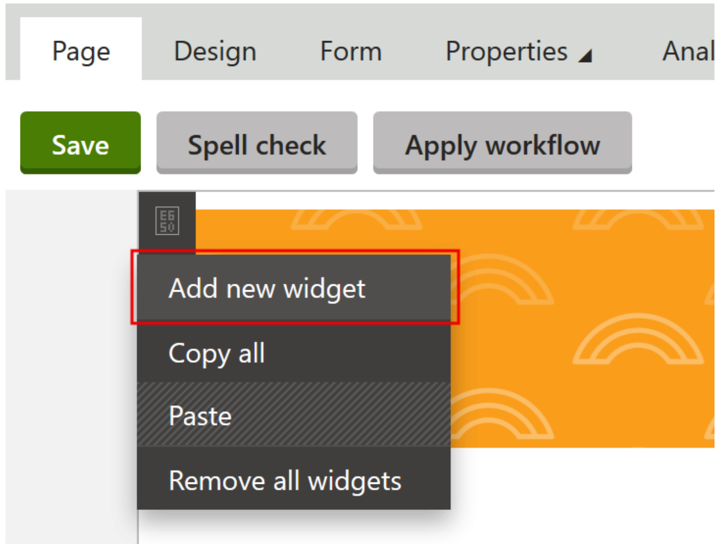
Widget zone type:

- None
- User personalization
- [✓ Customization by page editor]
- Customization by group administrator

Modular Designs - Widgets

Adding to a Page

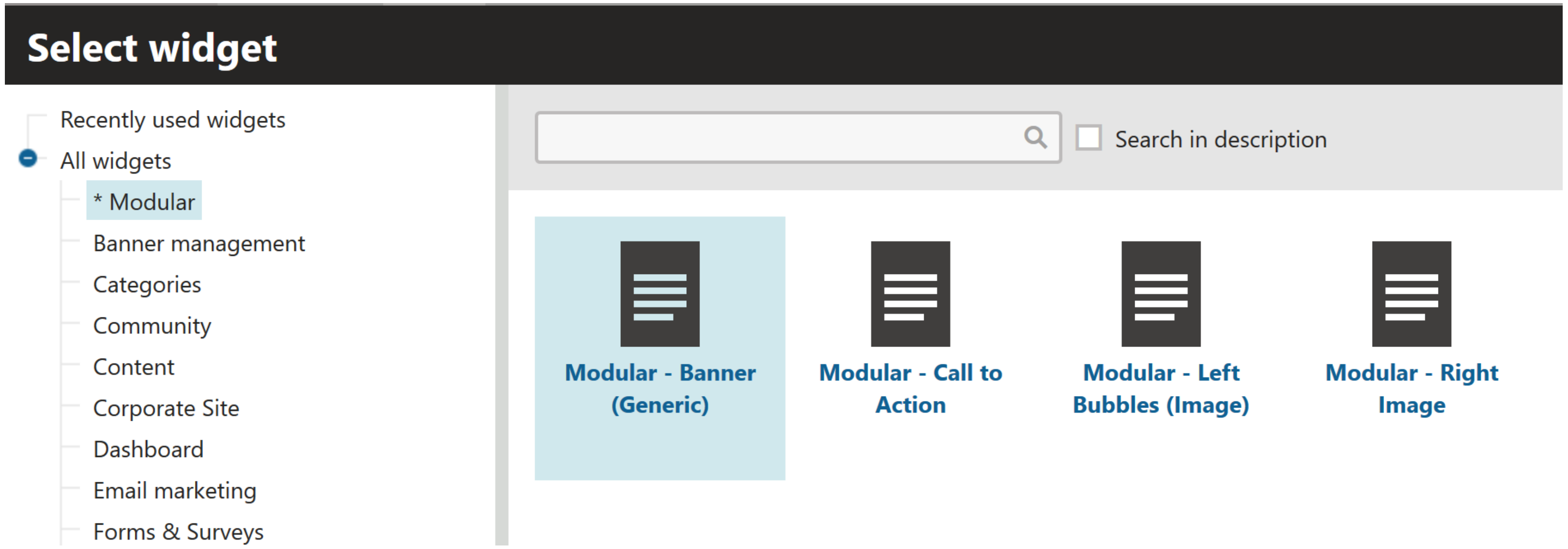
- The widgets can now be added on the page tab of any page that is using the page template just created.



Modular Designs - Widgets

Adding to a Page

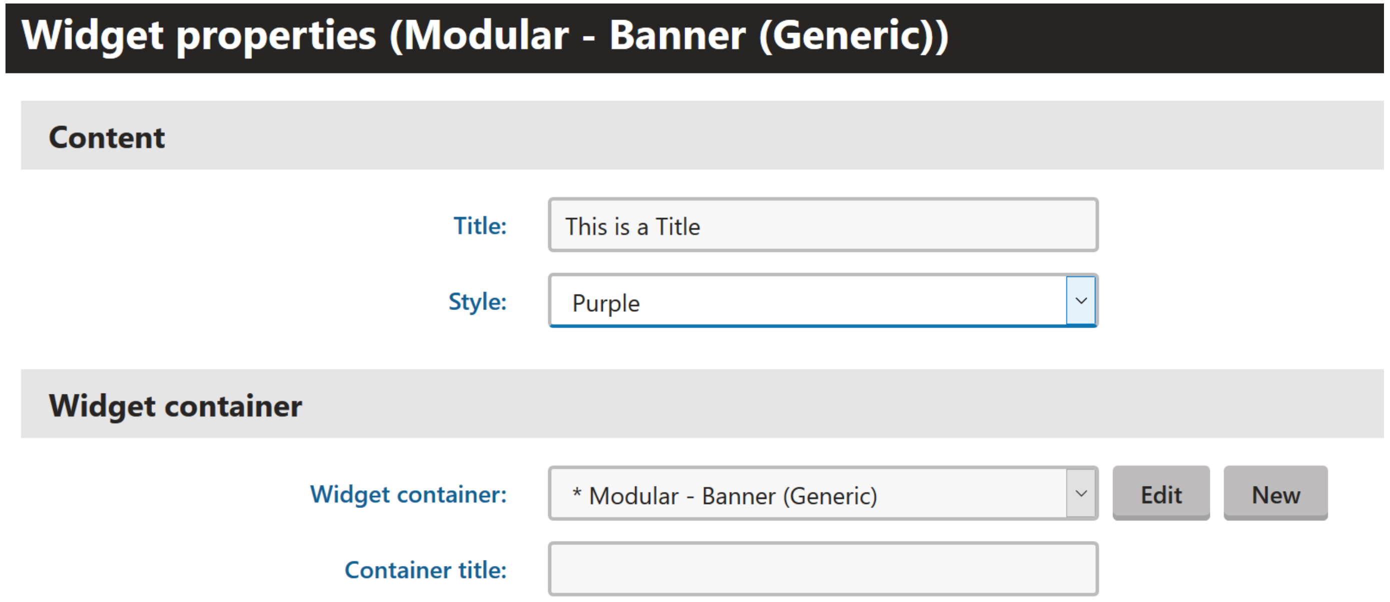
- Select the widget to add.



Modular Designs - Widgets

Adding to a Page

- Customize the Widget.



Widget properties (Modular - Banner (Generic))

Content

Title:	This is a Title
Style:	Purple

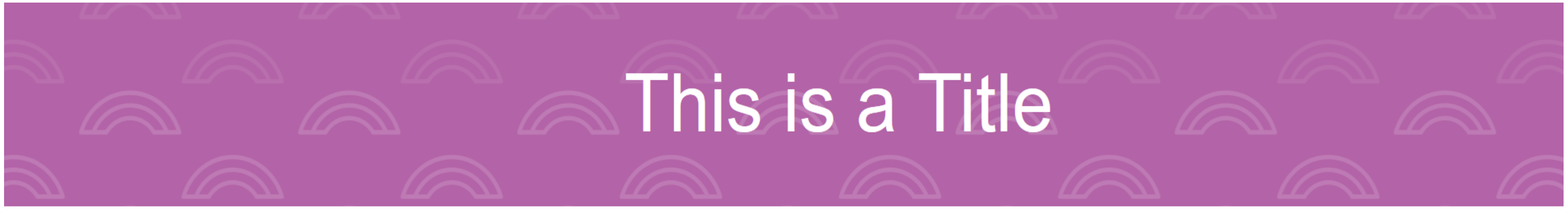
Widget container

Widget container:	* Modular - Banner (Generic)
Container title:	

Modular Designs - Widgets

Adding to a Page

- It now displays on the page.

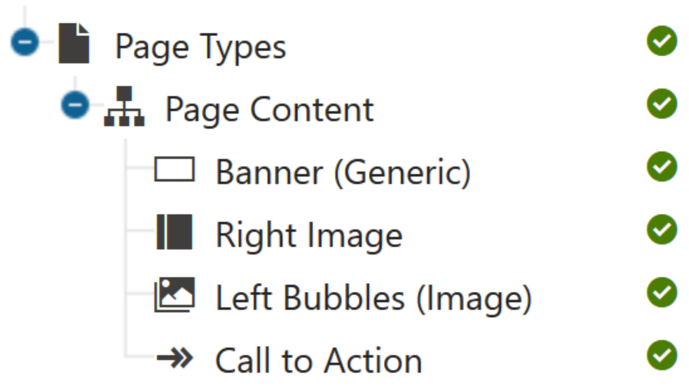


This is a Title

- Creating widgets like this allows the editor to visually build the page through the page tab.

Modular Designs – Page Types

Another approach is to use page types and the content tree instead of widgets.



The modular components of a page are built by adding items to the content tree as children of the current page.

Modular Designs – Page Types

Each page type only has one part.

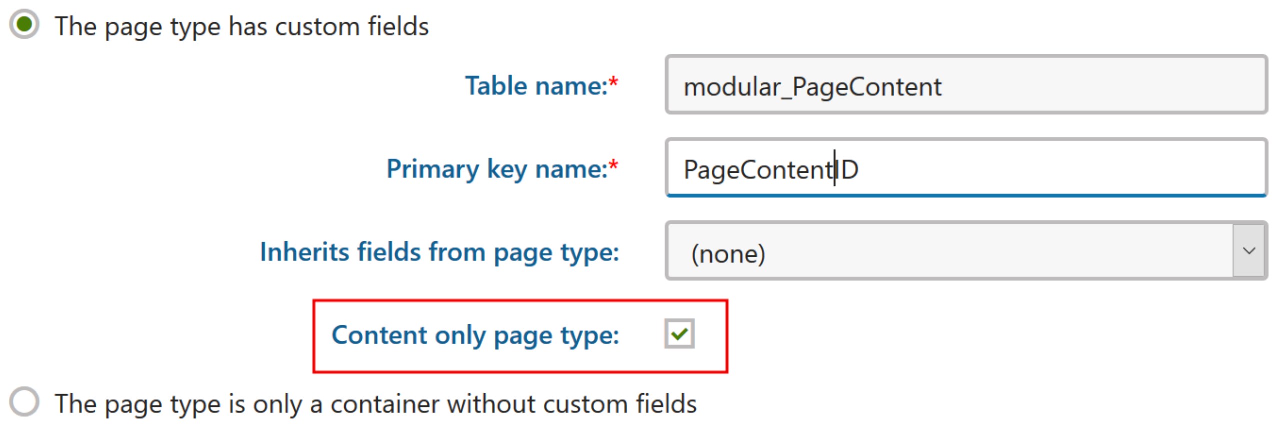
- Page Type – input fields and transformation (markup)

The page type’s transformation renders the markup.

Modular Designs – Page Types

Page Type – Page Content

- Modular page types are created in the Page types application within Kentico
- Create a page type to hold the other modular page types, name it **Page Content**
- Check the **Content only page type** box as these page types will never be pages



● The page type has custom fields

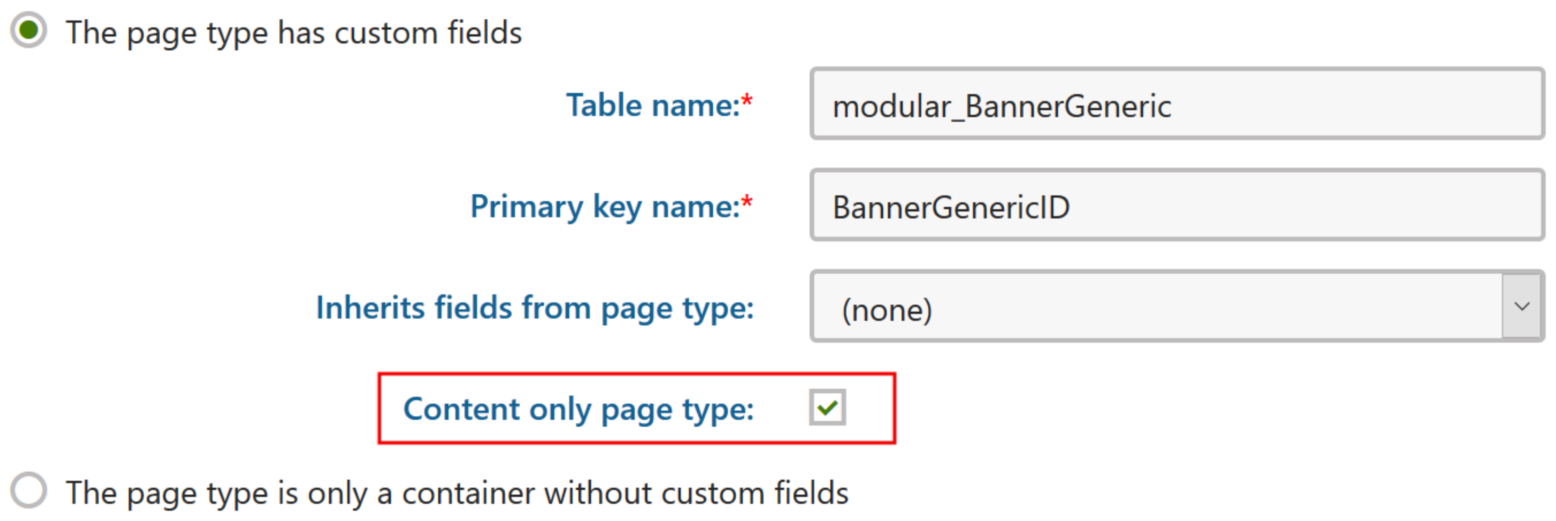
Table name:*	modular_PageContent
Primary key name:*	PageContentID
Inherits fields from page type:	(none)
Content only page type:	✓

○ The page type is only a container without custom fields

Modular Designs – Page Types

Modular Page Type

- Now create the modular page types for the page
- Also Check the **Content only page type** box for these page types



[● The page type has custom fields]

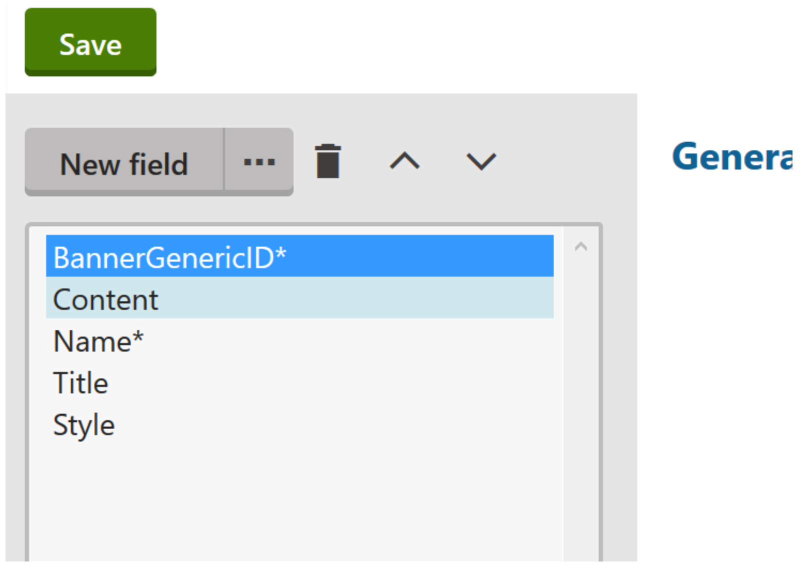
Table name:*	modular_BannerGeneric
Primary key name:*	BannerGenericID
Inherits fields from page type:	(none)
Content only page type:	✓

○ The page type is only a container without custom fields

Modular Designs - Page Types

Modular Page Type

- Add fields to the page type

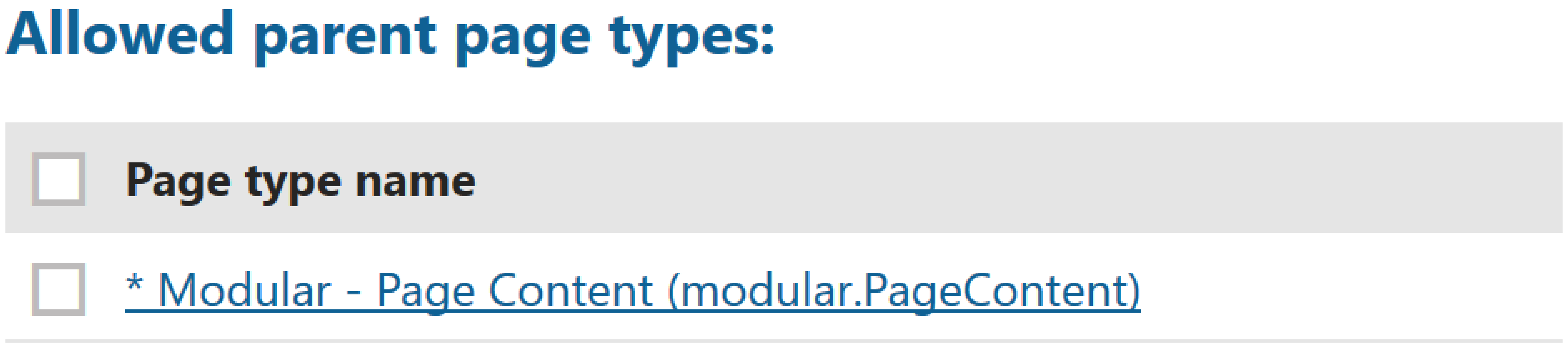


BannerGenericID*
Content
Name*
Title
Style

Modular Designs - Page Types

Modular Page Type

- Set the allowed parent page type to the Page Content page type created earlier



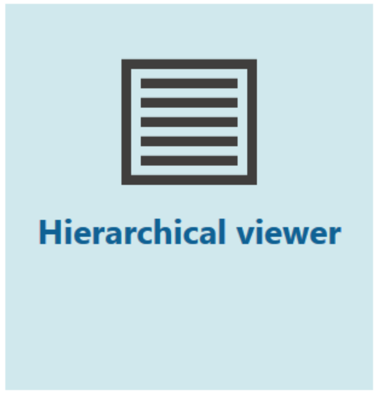
Allowed parent page types:

☐	Page type name
☐	* Modular - Page Content (modular.PageContent)

Modular Designs - Page Types

Page Template

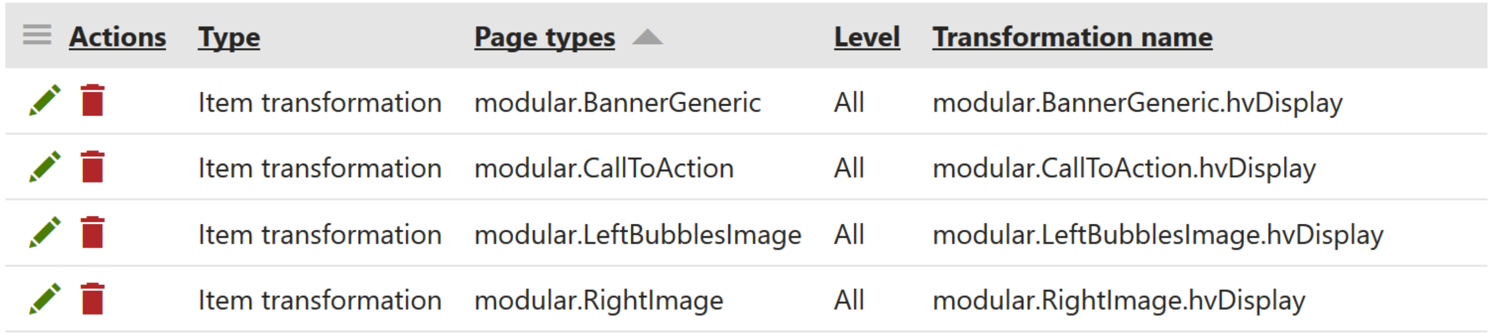
- On the page template that will display the modular page types, add a web part zone
- In the zone, add a Hierarchical viewer web part



Modular Designs - Page Types

Page Template

- The path for the viewer should be **./page-content/%**, the modular page content container for the page.
- Create a new hierarchical transformation to render the modular page types, each modular page type will have its own transformation so it can be rendered.

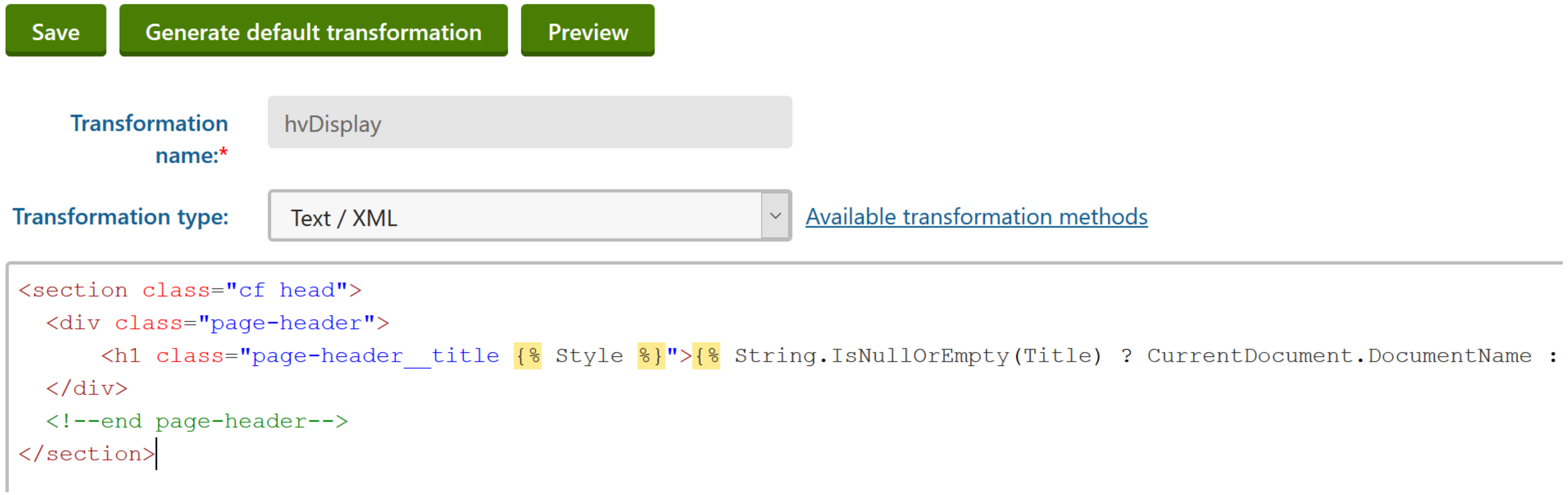


Actions	Type	Page types	Level	Transformation name
	Item transformation	modular.BannerGeneric	All	modular.BannerGeneric.hvDisplay
	Item transformation	modular.CallToAction	All	modular.CallToAction.hvDisplay
	Item transformation	modular.LeftBubblesImage	All	modular.LeftBubblesImage.hvDisplay
	Item transformation	modular.RightImage	All	modular.RightImage.hvDisplay

Modular Designs - Page Types

Page Template

- The markup for the transformation is the same as the web part container markup



Transformation name:* hvDisplay

Transformation type: Text / XML

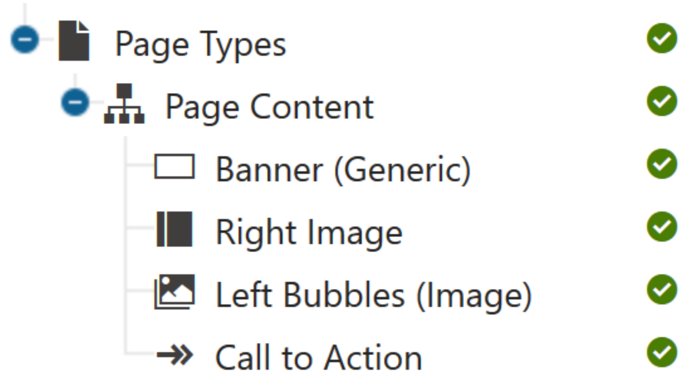
Available transformation methods

```
<section class="cf head">
  <div class="page-header">
      <h1 class="page-header__title {% Style %}">{% String.IsNullOrEmpty(Title) ? CurrentDocument.DocumentName :
  </div>
  <!--end page-header-->
</section>
```

Modular Designs – Page Types

Adding to a Page

- The modular page type can now be added to a page by adding a Page Content child and then adding the modular page types as children underneath Page Content.

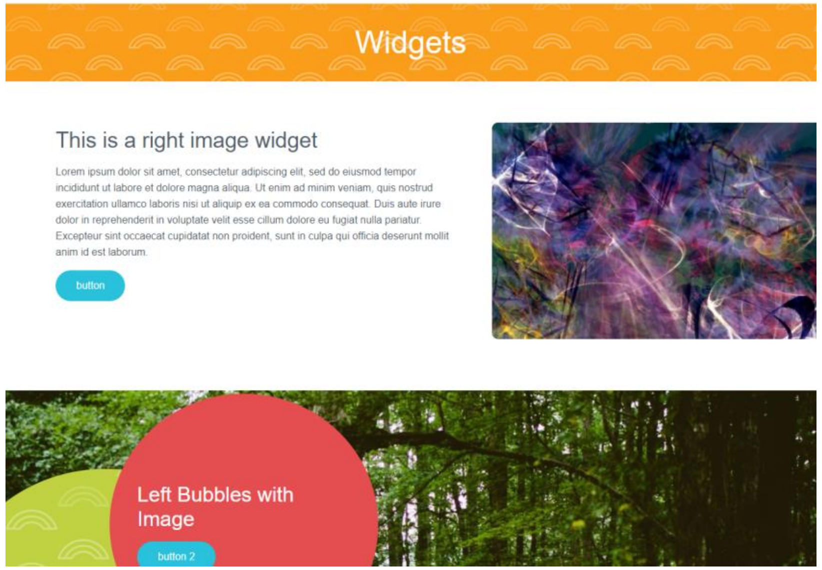


- Page Types ✓
  - Page Content ✓
    - Banner (Generic) ✓
    - Right Image ✓
    - Left Bubbles (Image) ✓
    - Call to Action ✓

Modular Designs

The rendered output for widgets and page types is identical

Widgets



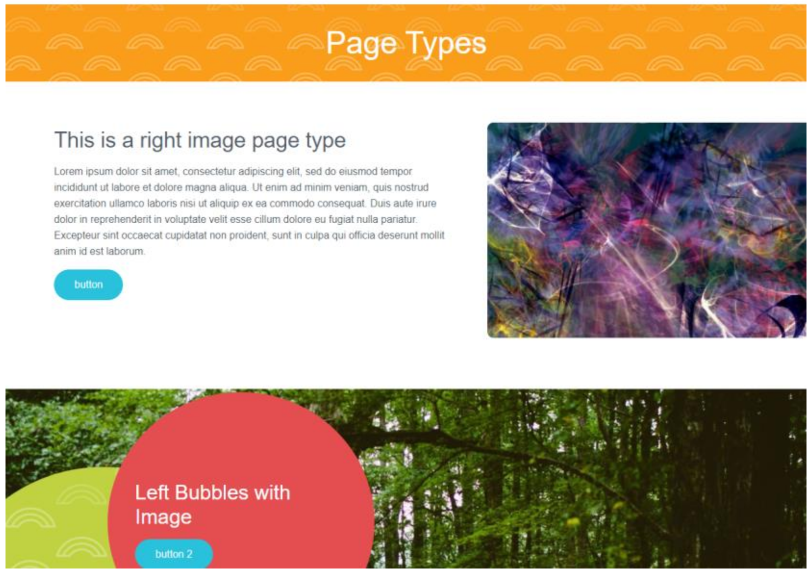
Widgets

This is a right image widget

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Left Bubbles with Image

Page Types



Page Types

This is a right image page type

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Left Bubbles with Image

Modular Designs – Discussion

For both approaches we create visual guides to help editors build pages using modular components

Pros and Cons

- So what are the advantages of one approach over the other?
- Are there certain situations when one should be used?

Modular Designs – Pros/Cons (Widgets)

Pros

- Can build the page visually on the form tab
- Personalization
- Easy to implement
- Content tree is clean

Cons

- Widgets don’t handle parent/child relationships very well
- Using web part containers in this way wasn’t what they were intended for

Modular Designs – Pros/Cons (Page Types)

Pros

- Content tree clearly displays what will be displayed on the page
- Handles parent/child relationship well using parent/child scopes
- Easy to implement

Cons

- Content tree is cluttered
- Not as visual as widgets
- Personalization not supported

Questions?

Joel Dahlin | joel@dahlindev.com | 320.281.0605

Jeff Steil | jeff@dahlindev.com | 320.492.9451

Dahlin Development | dahlindevelopment.com

* Follow our blog - Slides from this presentation, Kentico tips, advice, and more! *