

Creating and Applying Presentation



Module Objectives

By the end of this module, you will be able to:

- Describe how Sitecore layouts and components are tied to the ASP.NET MVC framework.
- Retrieve field values from Sitecore and make the fields editable in the Experience Editor.
- Explain the difference between inline-editable and non-editable field types.
- Apply presentation to a Sitecore item or Standard Values.
- Define the context item.
- Set up a control or component to target an item that is not the context item.
- Identify components in a HTML template.
- Describe the presentation details and their related definition items.





Creating Layouts

Presentation

Presentation Details

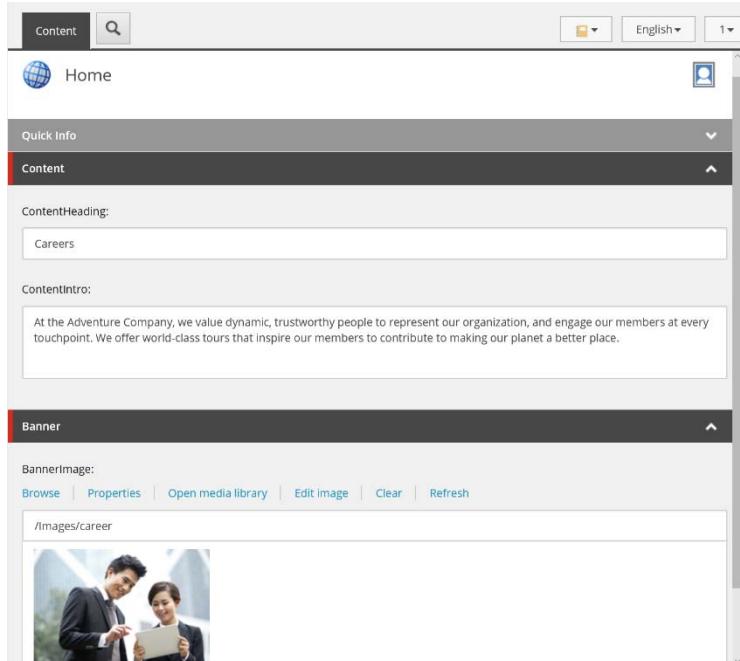
Devices

Anatomy of Layout



Creating Layouts

Presentation



Content

Home

Quick Info

Content

ContentHeading: Careers

ContentIntro: At the Adventure Company, we value dynamic, trustworthy people to represent our organization, and engage our members at every touchpoint. We offer world-class tours that inspire our members to contribute to making our planet a better place.

Banner

BannerImage:

Browse Properties Open media library Edit image Clear Refresh

/Images/career



THE ADVENTURE COMPANY HOME

Careers

Careers

At the Adventure Company, we value dynamic, trustworthy people to represent our organization, and engage our members at every touchpoint. We offer world-class tours that inspire our members to contribute to making our planet a better place.

Developer

The Adventure Company is currently looking for highly-skilled .NET developers to join the Product Development department's team in London and contribute to our vision of enabling our customers to own their Adventure Company experience.

HR Manager

The Adventure Company is looking for an experienced Regional HR Manager with first-hand knowledge of the travel industry, who is used to working with highly specialized and qualified colleagues.

Sales Support

The Sales Support team member is an expert in contract processing, and will team up with the global team for ensuring best practice and knowledge sharing in order to provide world-class sales support.

[Read more](#)

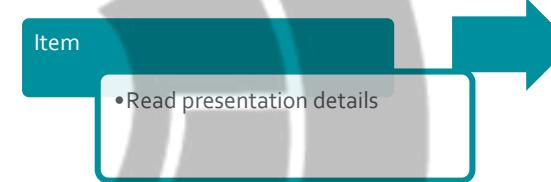
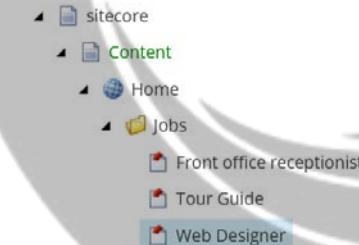
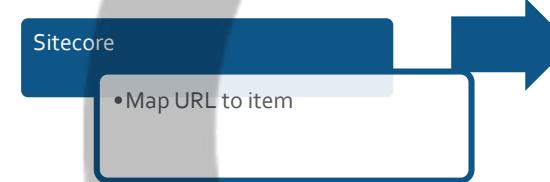
Support for Webforms or MVC

Presentation carried out by:

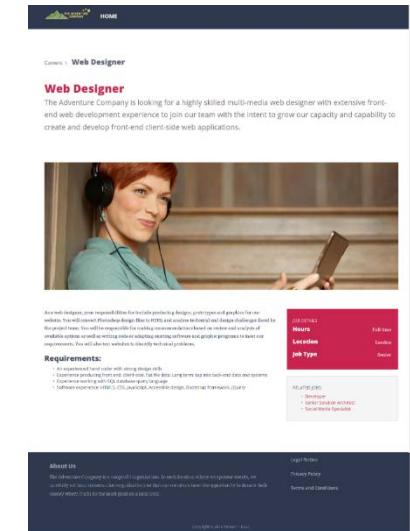
- Layout - Overall scaffold of the page
- Component - Generates mark up for a smaller part of the page



Property of items storing information Sitecore requires to build page response



ASP.NET



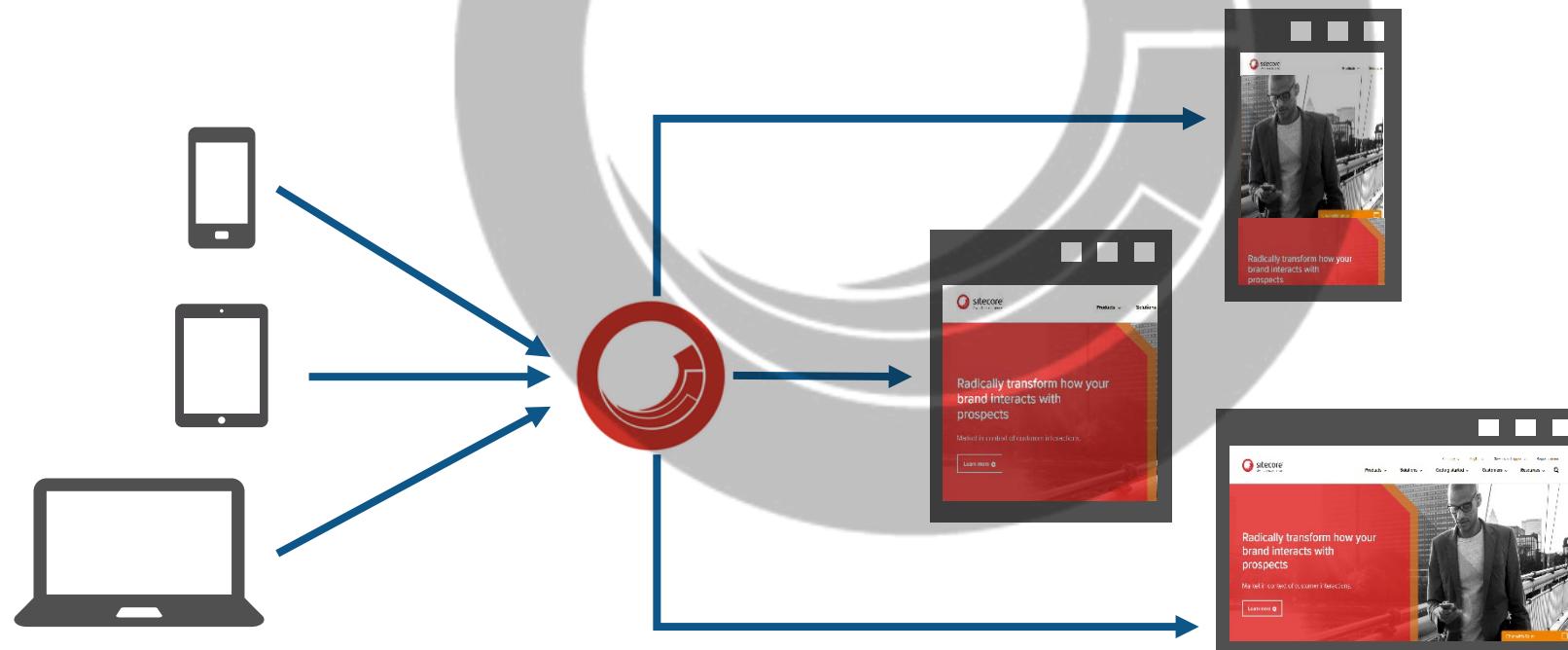
No presentation details set = no layout error

Best practice - set presentation details on Standard Values

Full support of responsive design

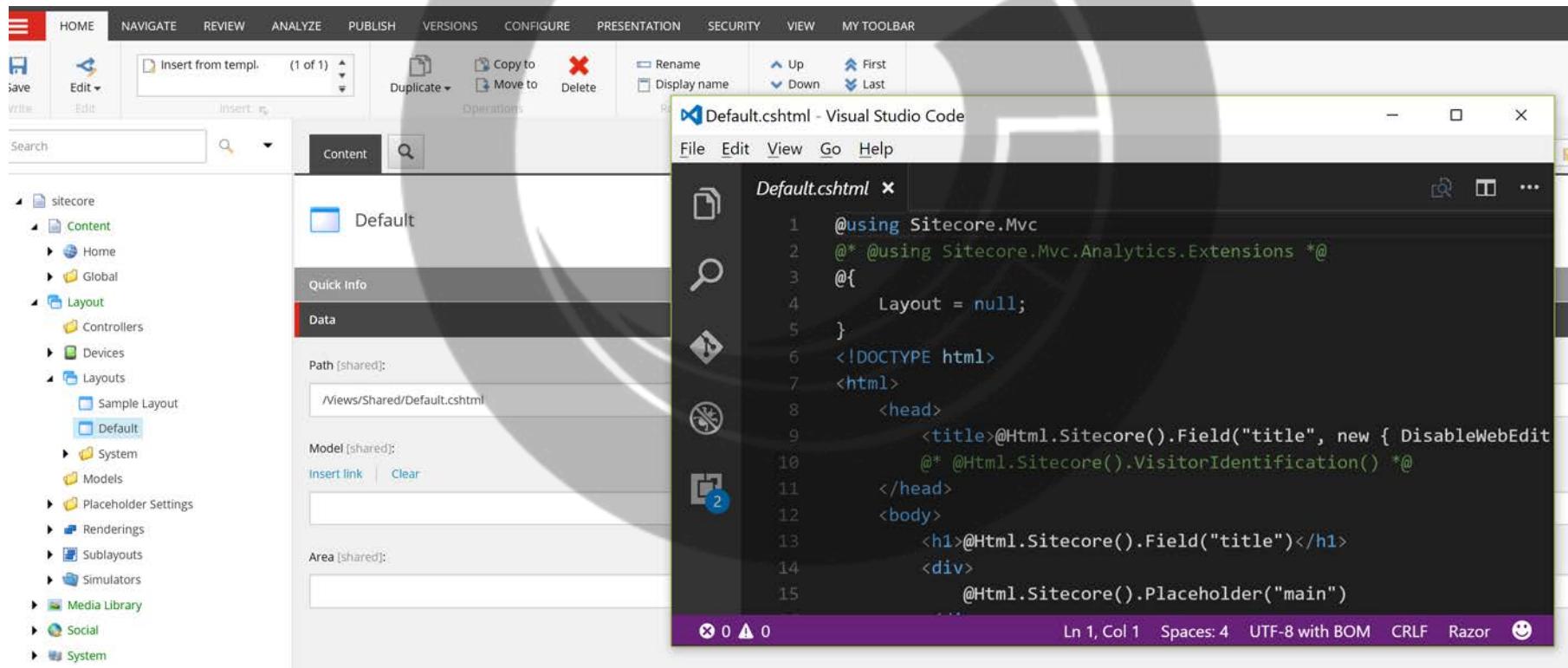
Adaptive design with devices

Devices provide alternate presentation detail configurations



.cshtml file and definition item

Path field links item and file



Demo: Setting Presentation Details

In this demo, you will see what happens when an item with no presentation details is requested. Then you will learn how to create a new layout and bind it in the presentation details.

Lab: Creating a Layout

In this lab, you will create layout and add the static HTML from the templates to it. You will then bind it on the presentation details so it gets used to render the page response.



In-page editing

Editing fields

Editing in the Experience Editor

Field Parameters

Image Manipulation

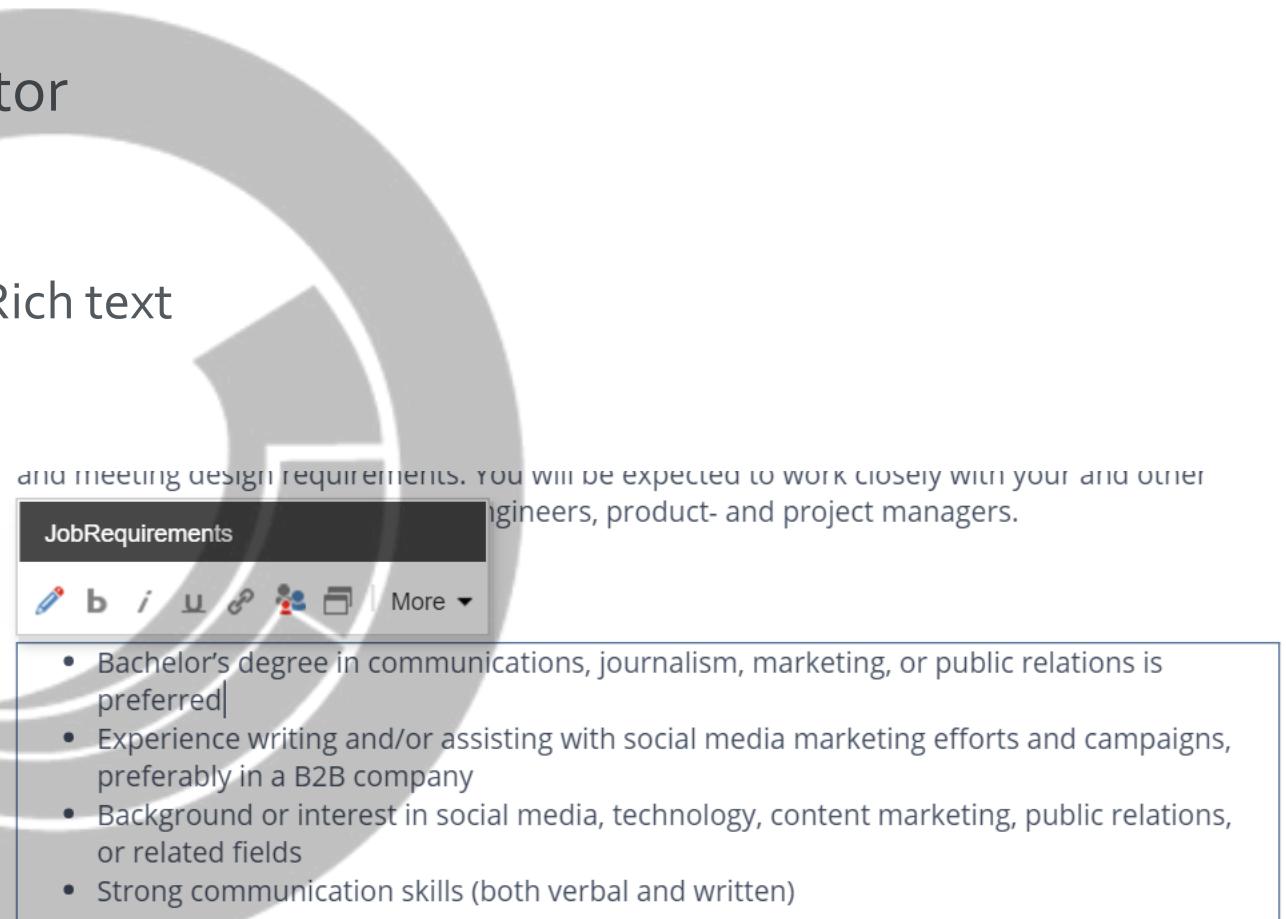
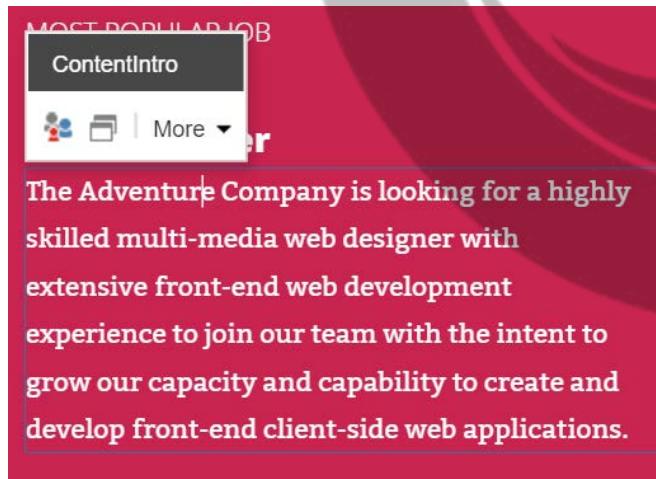


In-page editing

All fields can be edited in the Content Editor

Inline editing in Experience Editor for:

- Text fields - Single-Line Text, Multi-Line Text, Rich text
- Image
- Date and Datetime
- General Link field



The screenshot shows the Sitecore Content Editor interface. A modal dialog is open with the title "JobRequirements". Inside the dialog, there is a rich text editor toolbar with icons for bold, italic, underline, etc. Below the toolbar, the text "and meeting design requirements. You will be expected to work closely with your and other engineers, product- and project managers." is visible. At the bottom of the dialog, there is a list of bullet points:

- Bachelor's degree in communications, journalism, marketing, or public relations is preferred
- Experience writing and/or assisting with social media marketing efforts and campaigns, preferably in a B2B company
- Background or interest in social media, technology, content marketing, public relations, or related fields
- Strong communication skills (both verbal and written)

The Adventure Company is proud to be an equal opportunity workplace, and is an affirmative action employer. We are committed to equal employment opportunity regardless of race, color, ancestry, religion, gender, national origin, sexual orientation, age, citizenship, marital status, disability, veteran status or any other legally protected basis.

@Html.Sitecore().Field("name of the field")



```

```

Makes field editable inline

Renders clean HTML on live site

Works with all in-line editable field types

Force data to come from another item

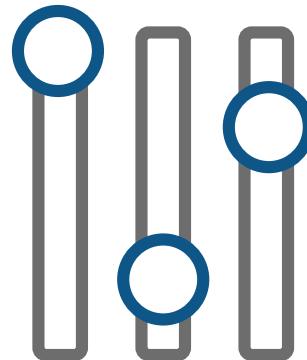
```
@Html.Sitecore().Field("myfield",anItemObject)
```

Add extra parameters through anonymous object

```
@Html.Sitecore().Field("myImageField", new {@class="imageright" })
```

Available parameters:

- DisableWebEdit - Boolean - Will disable inline editing in Experience Editor
- EnclosingTag - String - Surrounds rendering result in another tag
- Format - String - Formatting string for date and date time fields
- @class - String - Add a class attribute to or <a> tags



Resizing images on the server - cached
on file system

```
@Html.Sitecore().Field("my image  
field", new {mw=120, mh=120})
```

- mw - Maximum Width
- mh - Maximum Height
- w - Width
- h - Height
- as - Allow Stretch (allow changing aspect ratio)
- bc - Background Color (when aspect ratio changes
and image does not fill entire area)



After

Demo: Working with the Experience Editor

In this demo, you will see how to render the contents of a field and the different manipulation options available.

Lab: Making Content Editable in the Experience Editor

In this lab, you are going to start making the site editable. You will replace some of the hardcoded HTML for content stored in Sitecore. You will then test how content is editable inline using the Experience Editor.

Componentizing Pages

[Componentizing a Design](#)

[View Renderings](#)

[Statically Including a Component](#)

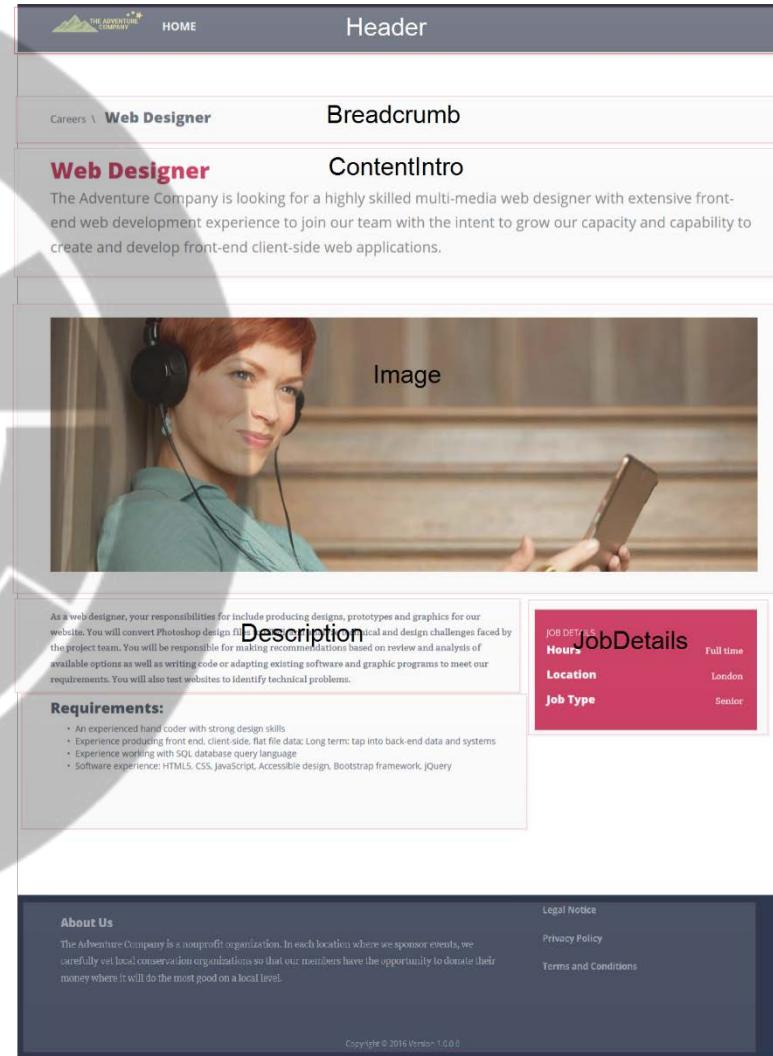


Componentizing Pages

The process of breaking down a design into discrete, re-usable, and generic components

When to create a separate component

- Is this HTML viewed on more than one page?
- Is there the chance that this HTML might be removed or moved by users?
- Does this HTML act as scaffolding?



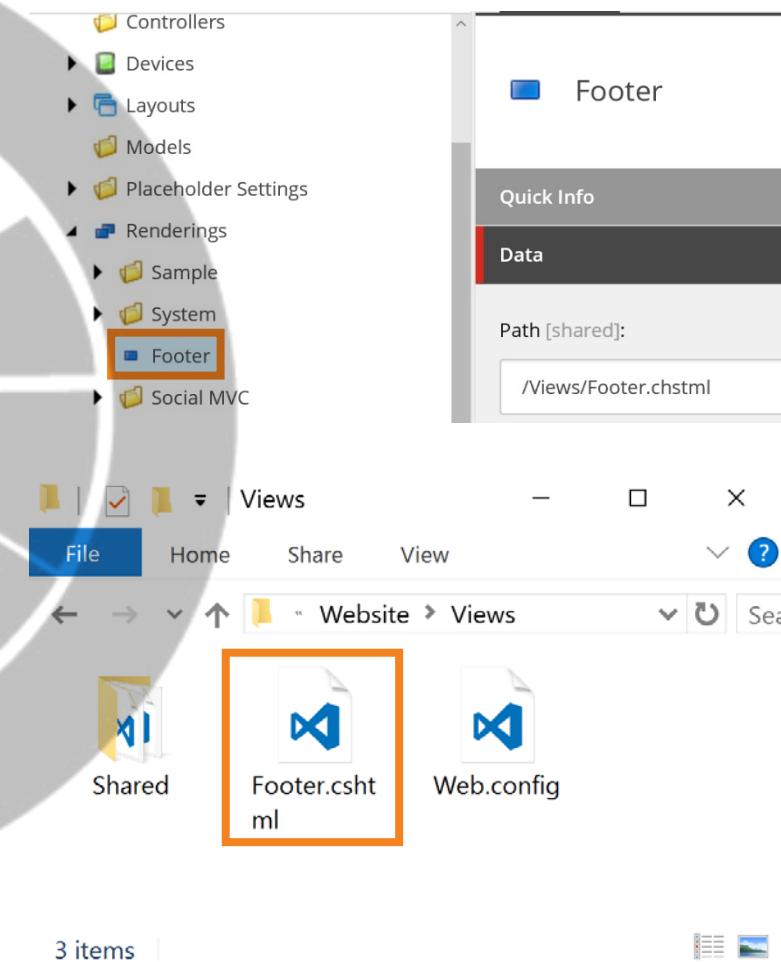
MVC Components: ViewRenderings & ControllerRenderings

ViewRendering

Razor .cshtml file

Need to register it in Sitecore with a definition item

No need to create Controller or Model

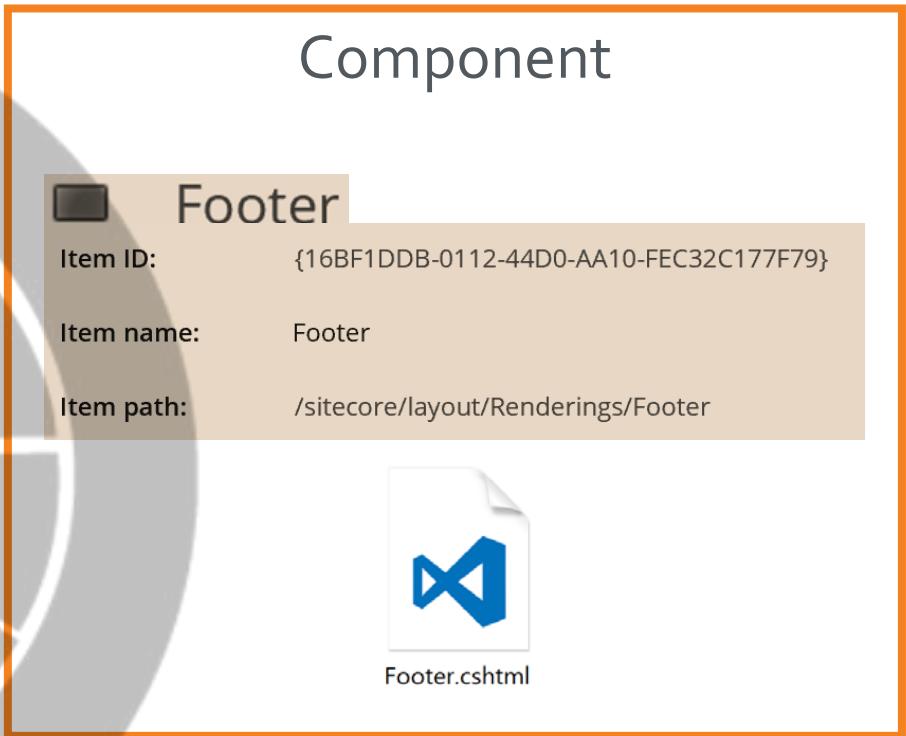


Statically Including a Component



Include a component inside another one

```
@Html.Sitecore().Rendering  
("path or id to definition item")
```



```
@Html.Sitecore().Rendering("{  
16BF1DDB-0112-44D0-AA10-  
FEC32C177F79}")
```

Demo: Creating a component

In this demo, you will see how to create a new View Rendering component and then add it to the page statically.

Lab: Componentizing the layout

In this lab, you will split the HTML from the layout into smaller, more manageable and reusable units: components. You will then statically reference those components on the layout so they are still shown on the page.

Dynamically build pages

Dynamic Binding

Using Placeholders

Placeholder Definition Items

Overriding and Resetting Presentation

One layout per device



Dynamically build pages

Ability to add components onto the page when building the response

How?

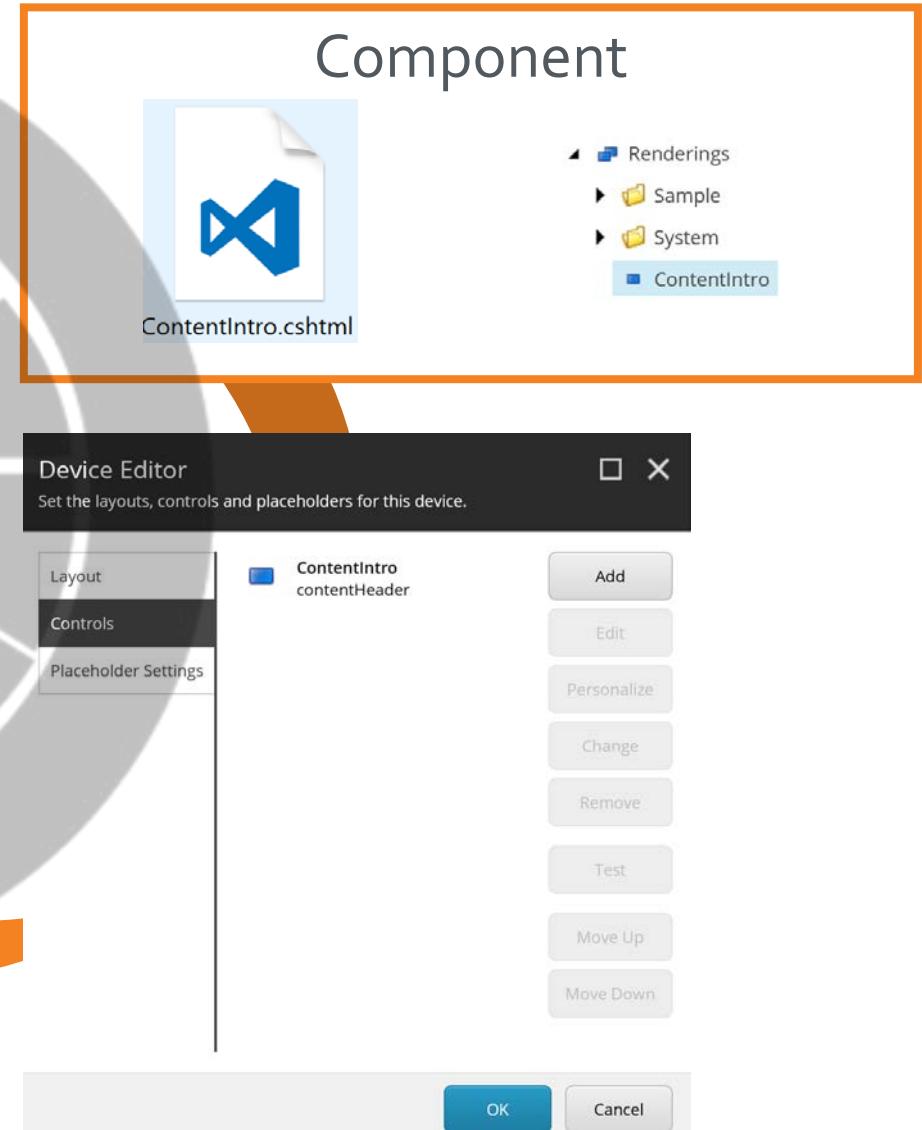
1. Add a placeholder to the markup
2. Add a mapping between component and placeholder on the Presentation Details



Using Placeholders

Add a placeholder to any .cshtml view/layout

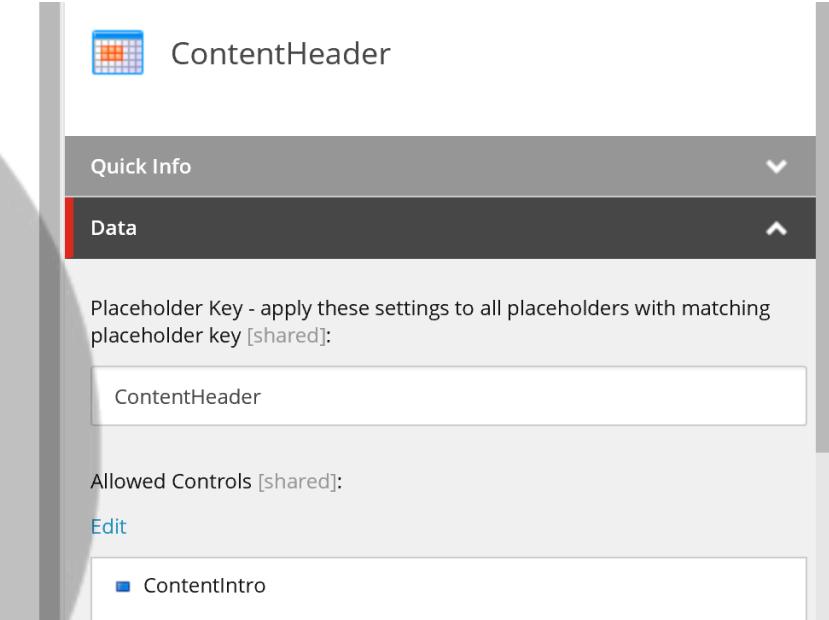
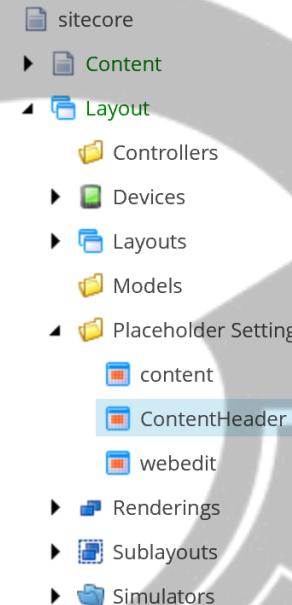
```
@Html.Sitecore().Placeholder("placeholder key")
```



Designing feature of Experience Editor
modifies Presentation Details

No Placeholder Settings - editors can't
add components to that placeholder

Placeholder Settings item - configures
behavior placeholder in Experience
Editor

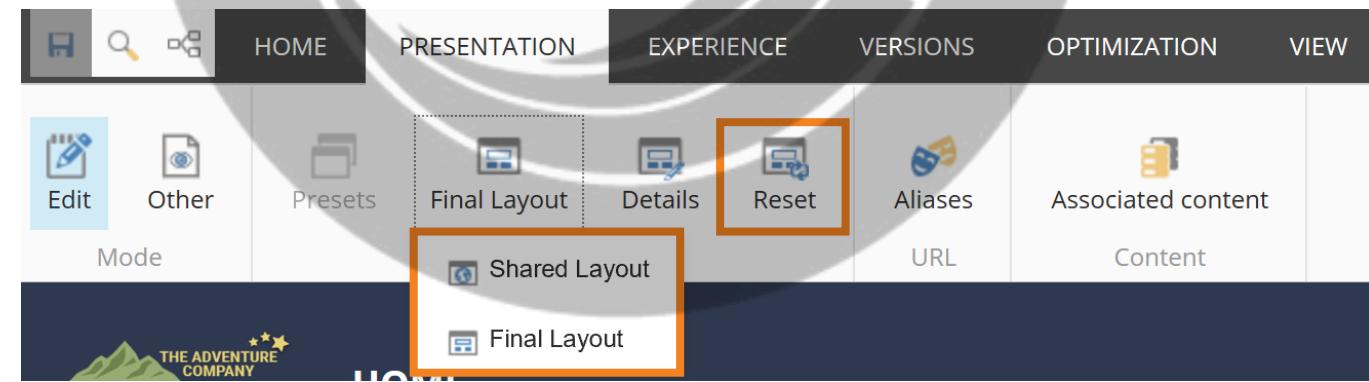


Developers usually set presentation details on Standard Values

Editors will override them on items

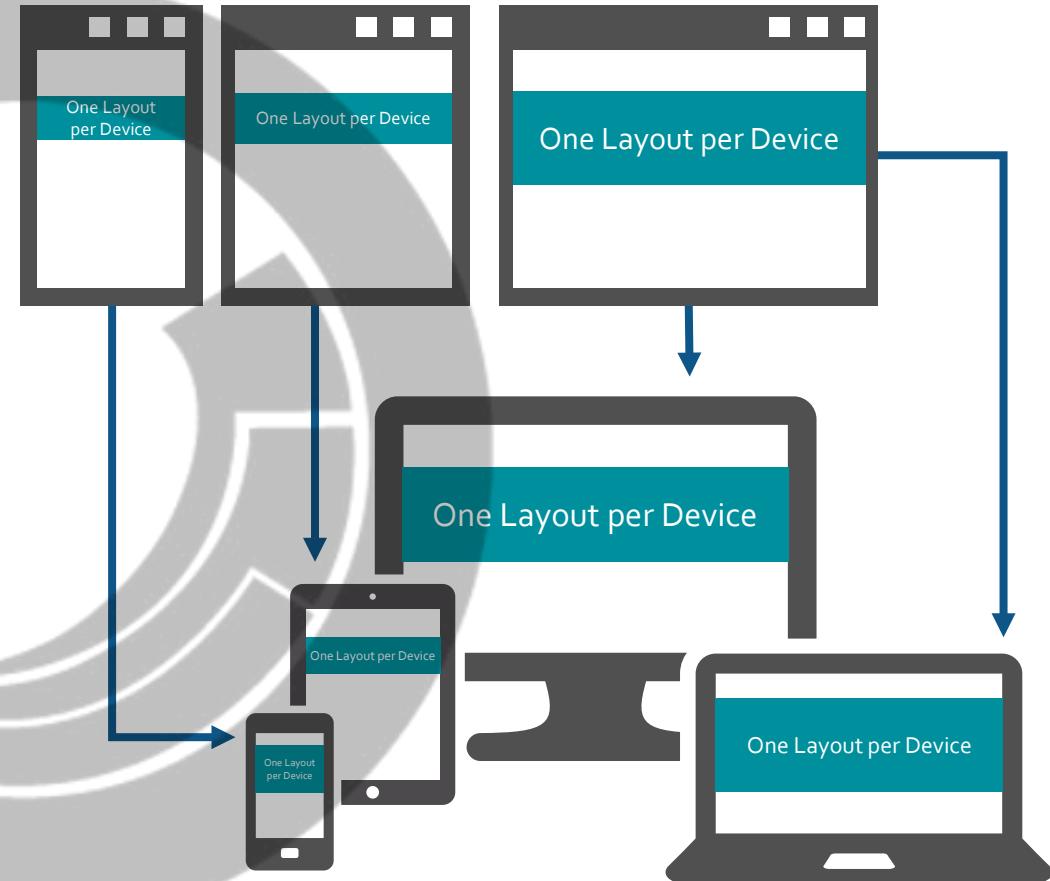
- Final - override on that version
- Shared - override on all versions in all languages

Possibility to reset back to Standard Values



Layout (common scaffolding)

- + Statically add components that are needed in all pages
- + Dynamic binding for components that vary between pages
- = single layout per device



Demo: Introducing placeholders and their definition items

In this demo, you will see how to inject a component dynamically using placeholders.

Lab: Adding placeholders and definition items

In this lab, you are going to continue to create components to splice the HTML in the layout. Because you will work with the part of the design that changes between the 3 different pages, you will have to create placeholder and use dynamic binding.

Lab: Assigning the components to the placeholders

In this lab, you will use the presentation details to assign components to placeholders.

This will enable you to reconstruct the original look and feel of the page.

Lab: Setting the allowed controls

In this lab, you are going to test what happens to your page when is viewed in the Experience Editor. In particular you will see how the designing features are severely limited unless you configure them through Placeholder Settings.

Optional Lab: Completing the JobList component

In this lab, you are going to remove all the static content from the JobList component. In order to do that you need to know how to do two things: get the children of an item and get the URL of an item.

Component Data Source

[Setting a Data Source](#)
[Reading a Data Source](#)



Component Data Source

Give editors ability to decide which item data comes from

Careers

Careers

At the Adventure Company, we value dynamic, trustworthy people to represent our organization, and engage our members at every touchpoint. We offer world-class tours that inspire our members to contribute to making our planet a better place.

Developer

The Adventure Company is currently looking for highly-skilled .NET developers to join the Product Development department's team in London and contribute to our vision of enabling our customers to own their Adventure Company experience.

HR Manager

The Adventure Company is looking for an experienced Regional HR Manager with first-hand knowledge of the travel industry, who is used to working with highly specialized and qualified colleagues.

Sales Support

The Sales Support team member is an expert in contract processing, and will team up with the global team for ensuring best practice and knowledge sharing in order to provide world-class sales support.

Web designer

The Adventure Company is looking for a highly skilled multi-media web designer with extensive front-end web development experience to join our team with the intent to grow our capacity and capability to create and develop front-end client-side web applications.

[Read more](#)

FeaturedJob

Usage: 1 | More ▾

Add associated content

Edit the related item

Web designer

The Adventure Company is looking for a highly skilled multi-media web designer with extensive front-end web development experience to join our team with the intent to grow our capacity and capability to create and develop front-end client-side web applications.

[Read more](#)

Field() method already reads the datasource.

Default model for ViewRendering is RenderingModel

- .Item - Datasource - or context item if datasource note set
- .PageItem - context item
- .Rendering - access to more advanced properties



Demo: Creating a component that accepts a data source

In this demo, you will see how set the
datasource of a component.

Lab: Applying a data source to an existing component

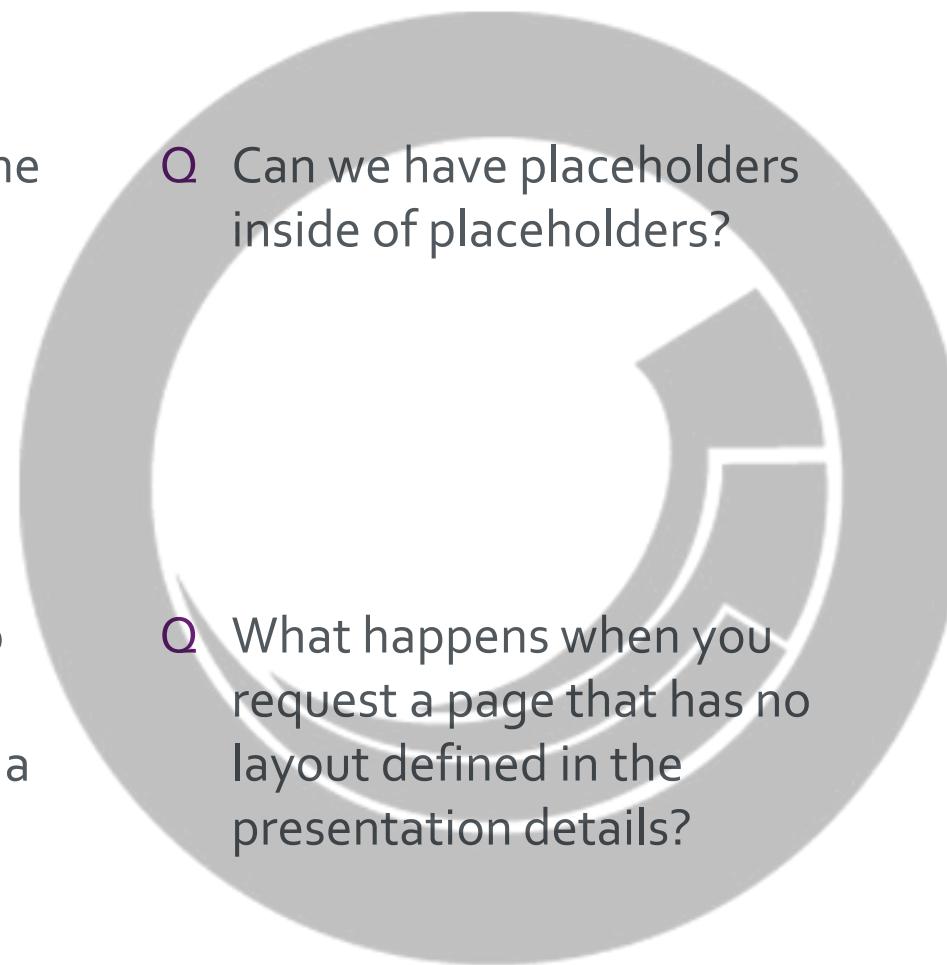
In this lab, you will set the datasource of a component using the Experience Editor. You will then modify the component's code so it uses that datasource to retrieve the content to be displayed.

Lab: Datasources in static components

In this lab, you will explore how datasources can be useful even in components where editors don't have the ability to change the component. You will use datasource to complete the footer component.

Q Why would I want to use the data source in a component?

Q What do you need to do to allow users to add and remove components from a placeholder using the Experience Editor?



Q Can we have placeholders inside of placeholders?

Q What happens when you request a page that has no layout defined in the presentation details?

Q Why is it a good idea to avoid using Rich Text fields as much as possible?

Q Why do you define Presentation Details on Standard Values if the Standard Values item is never going to be mapped to the requested URL?

Q Can you edit an item in the Experience Editor when the item has no presentation details set?

Q Which fields are editable inline using the Experience Editor?

Q Give an example of when you would use the DisableWebEdit parameter when rendering a field.



You have learned how to:

- Create a layout and bind it to the presentation details
- Render field content and make it editable
- How to split the presentation into smaller components
- Adding components statically and dynamically
- Using data sources