Home (/en/) / EPiServer (/en/episerver/75/episerver/) / 75 (/en/episerver/75/episerver/) / Properties

EPiServer 7.5 (/en/episerver/75/episerver/)

The EPiServer CMS Visual Studio Extension (/en/episerver/75/episerver-vs-extension/)

Properties (/en/episerver/75/properties/)

Property rendering (/en/episerver/75/property-rendering/)

Blocks (/en/episerver/75/blocks/)

Logging (/en/episerver/75/logging/)

Performance (/en/episerver/75/performance/) ›

Dynamic Data Store (/en/episerver/75/dynamic-data-store/)

Links (/en/episerver/75/links/)

Properties

April 21, 2014

You can use a lot of different properties in your page or block type. Here is a list.

```
// ------------------------------------------------
// PropertyBoolean
// ------------------------------------------------
[Display(
Name = "Show teaser",
Description = "",
GroupName = SystemTabNames.Content,
Order = 80)]
public virtual bool ShowTeaser { get; set; }



// ------------------------------------------------

// PropertyCategory
// ------------------------------------------------
[Display(
Name = "Selected categories",
Description = "",
GroupName = SystemTabNames.Content,
Order = 90)]
public virtual CategoryList SelectedCategories { get; set; }



```

```
// ---------------------------------------------------
// PropertyContentArea
// ---------------------------------------------------
[Display(
Name = "Right block area",
Description = "",
GroupName = SystemTabNames.Content,
Order = 70)]
public virtual ContentArea RightBlockArea { get; set; }



// ---------------------------------------------------
// PropertyContentReference (Image)
// Now using ContentReference for images.
// In EPiServer 7 we used URL
// ---------------------------------------------------
[UIHint(UIHint.Image)]
[Display(
Name = "Image",
Description = "",
GroupName = SystemTabNames.Content,
Order = 40)]
public virtual ContentReference Image { get; set; }



// ---------------------------------------------------
// PropertyContentReference (Document)
// Now using ContentReference for files.
// In EPiServer 7 we used URL
// ---------------------------------------------------
[UIHint(UIHint.MediaFile)]
[Display(
Name = "File",
Description = "",
GroupName = SystemTabNames.Content,
Order = 50)]
public virtual ContentReference File { get; set; }



// ---------------------------------------------------
// PropertyDate
// ---------------------------------------------------
[Display(
Name = "Event start time",
Description = "",
GroupName = SystemTabNames.Content,
Order = 100)]
public virtual DateTime EventStartTime { get; set; }



// ---------------------------------------------------
// PropertyFloatNumber
// ---------------------------------------------------
[Display(
Name = "Price",
Description = "",
GroupName = SystemTabNames.Content,
Order = 110)]
public virtual double Price { get; set; }



// ---------------------------------------------------
// PropertyLinkCollection
// ---------------------------------------------------
[Display(
Name = "Links",
Description = "",
GroupName = SystemTabNames.Content,
Order = 60)]
public virtual LinkItemCollection Links { get; set; }



// ---------------------------------------------
// PropertyLongString (PropertyString)
// ----------------------------------------------
[Display(
Name = "Heading",
Description = "",
GroupName = SystemTabNames.Content,
Order = 10)]
public virtual string Heading { get; set; }



// ---------------------------------------------
// PropertyLongString (PropertyString)
// The same as above except that this has the UIHint
// The UIHint will make you have a multiline textbox for the Editor
// If you want the line breaks to be rendered for the visitor,
// check: http://joelabrahamsson.com/episerver-editing-delight-challenge-mvc-solut
// ----------------------------------------------
[UIHint(UIHint.Textarea)]
[Display(
Name = "Intro",
Description = "",
GroupName = SystemTabNames.Content,
Order = 15)]
public virtual string MainIntro { get; set; }



// ----------------------------------------------
// PropertyNumber
// ----------------------------------------------
[Display(
Name = "Votes",
Description = "",
GroupName = SystemTabNames.Content,
Order = 120)]
public virtual int Votes { get; set; }



// ----------------------------------------------
// PropertyNumber
// This example can be used to set how content
// should be sorted
// ----------------------------------------------
[BackingType(typeof(PropertyNumber))]
[UIHint("SortOrder")]
[Display(
GroupName = SystemTabNames.Content,
Order = 150)]
[DefaultValue(FilterSortOrder.PublishedDescending)]
public virtual FilterSortOrder SortOrder { get; set; }



// ----------------------------------------------
// PropertyPageReference
// ----------------------------------------------
[Display(
Name = "Sitemap page",
Description = "",
GroupName = SystemTabNames.Content,
Order = 55)]
public virtual PageReference SitemapPageLink { get; set; }



// ------------------------------------------------
// PropertyPageType
// ------------------------------------------------
[Display(
Name = "Filter for page type",
Description = "",
GroupName = SystemTabNames.Content,
Order = 130)]
public virtual PageType FilterPageType { get; set; }



// ------------------------------------------------
// PropertyString
// Only use if you really have to.
// Use PropertyLongString instead with [StringLength(255)]
// ------------------------------------------------
[BackingType(typeof(PropertyString))]
[Display(
Name = "Short intro",
Description = "",
GroupName = SystemTabNames.Content,
Order = 170)]
public virtual string ShortIntro { get; set; }



// ------------------------------------------------
// PropertyUrl
// ------------------------------------------------
[BackingType(typeof(PropertyUrl))]
[Display(
Name = "Link",
Description = "",
GroupName = SystemTabNames.Content,
Order = 30)]
public virtual Url Link { get; set; }



// ------------------------------------------------
// PropertyXForm
// ------------------------------------------------
[Display(
Name = "Form",
Description = "",
GroupName = SystemTabNames.Content,
Order = 140)]
public virtual XForm Form { get; set; }



// ------------------------------------------------
// PropertyXhtmlString
// ------------------------------------------------
[Display(
Name = "Main body",
Description = "",
GroupName = SystemTabNames.Content,
Order = 20)]
public virtual XhtmlString MainBody { get; set; }



```

```


// ---------------------------------------------------
// Custom property in Alloy MVC
// Custom properties will be described later
// ---------------------------------------------------
[BackingType(typeof(PropertyStringList))]
[Display(Order = 160)]
[UIHint(Global.SiteUIHints.Strings)]
[CultureSpecific]
public virtual string[] UniqueSellingPoints { get; set; }



// ---------------------------------------------------
// Dropdown
//  SelectOne needs a reference to EPiServer.UI and is in the EPiServer.Shell.Obje
// ---------------------------------------------------
[SelectOne(SelectionFactoryType = typeof(LanguageSelectionFactory))]
[Display(
GroupName = SystemTabNames.Content,
Order = 200)]
public virtual string SingleLanguage { get; set; }



// ---------------------------------------------------
// Checkbox list
// ---------------------------------------------------
[SelectMany(SelectionFactoryType = typeof(LanguageSelectionFactory))]
[Display(
GroupName = SystemTabNames.Content,
Order = 210)]
public virtual string MultipleLanguage { get; set; }

public class LanguageSelectionFactory : ISelectionFactory
{
    public IEnumerable GetSelections(ExtendedMetadata metadata)
    {
        return new ISelectItem[] { new SelectItem() { Text = "English", Value = "E
    }
}
```

This list was inspired by an article by Alexander Haneng (http://world.episerver.com/Blogs/Alexander-Haneng/Dates/2012/7/How-to-define-properties-in-EPiServer-7---A-quick-reference/). If you want to read more about properties you can also buy the book EPiServer 7 CMS Development (https://leanpub.com/episerver-7-cms-development) (written by Joel Abrahamsson).

So now you know how to define the properties. The next step is to know how to render/use the EPiServer properties (/en/episerver/75/property-rendering/).

About

This site will focus on EPiServer 7.5 with MVC. You will also find some information about Web Forms and older EPiServer versions.

Contact Us

Email: info@epiknowledge.com (mailto:info@epiknowledge.com)