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; }


```

```
26  // ------------------------------------------------
27  // PropertyContentArea
28  // ------------------------------------------------
29  [Display(
30  Name = "Right block area",
31  Description = "",
32  GroupName = SystemTabNames.Content,
33  Order = 70)]
34  public virtual ContentArea RightBlockArea { get; set; }
35
36
37
38  // ------------------------------------------------
39  // PropertyContentReference (Image)
40  // Now using ContentReference for images.
41  // In EPiServer 7 we used URL
42  // ------------------------------------------------
43  [UIHint(UIHint.Image)]
44  [Display(
45  Name = "Image",
46  Description = "",
47  GroupName = SystemTabNames.Content,
48  Order = 40)]
49  public virtual ContentReference Image { get; set; }
50
51
52
53  // ------------------------------------------------
54  // PropertyContentReference (Document)
55  // Now using ContentReference for files.
56  // In EPiServer 7 we used URL
57  // ------------------------------------------------
58  [UIHint(UIHint.MediaFile)]
59  [Display(
60  Name = "File",
61  Description = "",
62  GroupName = SystemTabNames.Content,
63  Order = 50)]
64  public virtual ContentReference File { get; set; }
65
66
67
68  // ------------------------------------------------
69  // PropertyDate
70  // ------------------------------------------------
71  [Display(
72  Name = "Event start time",
73  Description = "",
74  GroupName = SystemTabNames.Content,
75  Order = 100)]
76  public virtual DateTime EventStartTime { get; set; }
77
78
79
80  // ------------------------------------------------
81  // PropertyFloatNumber
82  // ------------------------------------------------
83  [Display(
84  Name = "Price",
85  Description = "",
86  GroupName = SystemTabNames.Content,
87  Order = 110)]
88  public virtual double Price { get; set; }
89
90
91
92  // ------------------------------------------------
93  // PropertyLinkCollection
94  // ------------------------------------------------
95  [Display(
```

```
96  Name = "Links",
97  Description = "",
98  GroupName = SystemTabNames.Content,
99  Order = 60)]
100 public virtual LinkItemCollection Links { get; set; }
101
102
103
104 // ------------------------------------------------
105 // PropertyLongString (PropertyString)
106 // ------------------------------------------------
107 [Display(
108 Name = "Heading",
109 Description = "",
110 GroupName = SystemTabNames.Content,
111 Order = 10)]
112 public virtual string Heading { get; set; }
113
114
115
116 // ------------------------------------------------
117 // PropertyLongString (PropertyString)
118 // The same as above except that this has the UIHint
119 // The UIHint will make you have a multiline textbox for the Editor
120 // If you want the line breaks to be rendered for the visitor,
121 // check: http://joelabrahamsson.com/episerver-editing-delight-challenge-mvc-solut
122 // ------------------------------------------------
123 [UIHint(UIHint.Textarea)]
124 [Display(
125 Name = "Intro",
126 Description = "",
127 GroupName = SystemTabNames.Content,
128 Order = 15)]
129 public virtual string MainIntro { get; set; }
130
131
132
133 // ------------------------------------------------
134 // PropertyNumber
135 // ------------------------------------------------
136 [Display(
137 Name = "Votes",
138 Description = "",
139 GroupName = SystemTabNames.Content,
140 Order = 120)]
141 public virtual int Votes { get; set; }
142
143
144
145 // ------------------------------------------------
146 // PropertyNumber
147 // This example can be used to set how content
148 // should be sorted
149 // ------------------------------------------------
150 [BackingType(typeof(PropertyNumber))]
151 [UIHint("SortOrder")]
152 [Display(
153 GroupName = SystemTabNames.Content,
154 Order = 150)]
155 [DefaultValue(FilterSortOrder.PublishedDescending)]
156 public virtual FilterSortOrder SortOrder { get; set; }
157
158
159
160 // ------------------------------------------------
161 // PropertyPageReference
162 // ------------------------------------------------
163 [Display(
164 Name = "Sitemap page",
165 Description = "",
```

```
166   GroupName = SystemTabNames.Content,
167   Order = 55)]
168   public virtual PageReference SitemapPageLink { get; set; }
169
170
171
172   // ------------------------------------------------
173   // PropertyPageType
174   // ------------------------------------------------
175   [Display(
176   Name = "Filter for page type",
177   Description = "",
178   GroupName = SystemTabNames.Content,
179   Order = 130)]
180   public virtual PageType FilterPageType { get; set; }
181
182
183
184   // ------------------------------------------------
185   // PropertyString
186   // Only use if you really have to.
187   // Use PropertyLongString instead with [StringLength(255)]
188   // ------------------------------------------------
189   [BackingType(typeof(PropertyString))]
190   [Display(
191   Name = "Short intro",
192   Description = "",
193   GroupName = SystemTabNames.Content,
194   Order = 170)]
195   public virtual string ShortIntro { get; set; }
196
197
198
199   // ------------------------------------------------
200   // PropertyUrl
201   // ------------------------------------------------
202   [BackingType(typeof(PropertyUrl))]
203   [Display(
204   Name = "Link",
205   Description = "",
206   GroupName = SystemTabNames.Content,
207   Order = 30)]
208   public virtual Url Link { get; set; }
209
210
211
212   // ------------------------------------------------
213   // PropertyXForm
214   // ------------------------------------------------
215   [Display(
216   Name = "Form",
217   Description = "",
218   GroupName = SystemTabNames.Content,
219   Order = 140)]
220   public virtual XForm Form { get; set; }
221
222
223
224   // ------------------------------------------------
225   // PropertyXhtmlString
226   // ------------------------------------------------
227   [Display(
228   Name = "Main body",
229   Description = "",
230   GroupName = SystemTabNames.Content,
231   Order = 20)]
232   public virtual XhtmlString MainBody { get; set; }
233
234
235
```

```
236
237
238   // ------------------------------------------------
239   // Custom property in Alloy MVC
240   // Custom properties will be described later
241   // ------------------------------------------------
242   [BackingType(typeof(PropertyStringList))]
243   [Display(Order = 160)]
244   [UIHint(Global.SiteUIHints.Strings)]
245   [CultureSpecific]
246   public virtual string[] UniqueSellingPoints { get; set; }
247
248
249
250   // ------------------------------------------------
251   // Dropdown
252   //  SelectOne needs a reference to EPiServer.UI and is in the EPiServer.Shell.Obje
253   // ------------------------------------------------
254   [SelectOne(SelectionFactoryType = typeof(LanguageSelectionFactory))]
255   [Display(
256   GroupName = SystemTabNames.Content,
257   Order = 200)]
258   public virtual string SingleLanguage { get; set; }
259
260
261
262   // ------------------------------------------------
263   // Checkbox list
264   // ------------------------------------------------
265   [SelectMany(SelectionFactoryType = typeof(LanguageSelectionFactory))]
266   [Display(
267   GroupName = SystemTabNames.Content,
268   Order = 210)]
269   public virtual string MultipleLanguage { get; set; }
270
271   public class LanguageSelectionFactory : ISelectionFactory
272   {
273       public IEnumerable GetSelections(ExtendedMetadata metadata)
274       {
275           return new ISelectItem[] { new SelectItem() { Text = "English", Value = "E
276       }
277   }
```

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)