G+1 3   Mer   Nästa blogg»   Skapa en blogg   Logga in

Na

WORK HERE   NEWS   OUR STORIES   CONTACT US

Stockholm   Our work   How we do it   About us
Chicago
London
New York

Monday, October 27, 2014

Page tree icons in EPiServer CMS 7.5

One appreciated feature we had in our old EPiServer 6 framework here at Nansen was the ability to set different icons for different page types in the page tree. This makes it easier for editors to distinguish between different page types to generally make

their life easier. And we all know the happier the editor, the better the content, and better content make for more visitors, right?

Anyways, the EPi 6 system we built was a bit of a hack, we added a ControlAdapter to the PageTree control to inject css classes on the different nodes depending on what page type it was.

Now in EPiServer 7.5 there are systems in place to make this a lot cleaner, using the tools provided by the CMS UI we can avoid using hacks, which make for a cleaner approach that will hopefully not break as easily should the underlying structure change in future updates.

CMS6 pagetree with custom icons

The most basic way is to add a editor descriptor for your class, like so:

```
1  [UIDescriptorRegistration]
2   public class ContainerPageUIDescriptor : UIDescriptor<containerpage>
3   {
4    public ContainerPageUIDescriptor()
5     : base(ContentTypeCssClassNames.Container)
6    {
7     DefaultView = CmsViewNames.AllPropertiesView;
8    }
9   }
10  </containerpage>
```

Now this approach will get tedious once you start adding page types as you will need one UI descriptor for each page type. There are several ways to make this more streamlined. This post will show you one approach using attributes on your page type classes.

We are going to use the EPiServer.Shell.UIDescriptor objects registered for all content types, these are added to the EPiServer.Shell.UIDescriptorRegistry during site startup. The UIDescriptor contain one important property that makes all of this work, IconClass. This class is added to the icon shown for each item in the page tree.

Requirements
I used EPiServer CMS v7.14.1 and EPiServer CMS UI version 7.15.0 when building this.

The parts
We will need to glue together three different parts to achieve this:

1. An attribute for page (content) type classes, where a icon class can be chosen
2. An initialization module to add the icon class to the page types
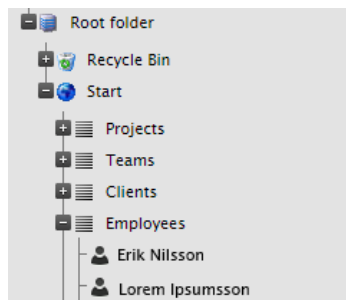3. Some icons

1. The attribute
The attribute is pretty straight forward, all we need is the ability to select a css class for our content types:

```
1  [AttributeUsage(AttributeTargets.Class)]
2  public class ContentIconAttribute : Attribute
3  {
4   public ContentIconAttribute(string iconClass)
5   {
```

Archive

► 2015 (15)
▼ 2014 (33)
　► December (2)
　► November (1)
　▼ October (1)
　　Page tree icons in EPiServer CMS 7.5
　► September (2)
　► June (6)
　► May (1)
　► April (15)
　► March (3)
　► January (2)
► 2013 (40)
► 2012 (44)
► 2011 (141)
► 2010 (106)
► 2009 (283)
► 2008 (133)

Labels

Nansen (50)
Web 2.0 expo (37)
Google (26)
microsoft (26)
Twitter (21)
Söderhavet (18)
ASP.NET (16)
Skeppsbron Mighty Penguins (14)
episerverdagen (13)
FOWA (12)
Internet (12)
Episerver Community 3 (9)
Google Analytics (9)
EPiServer CMS 6 (8)
Visual Studio (8)
jQuery (8)

```
 6      IconClass = iconClass;
 7    }
 8
 9    /// <summary>
10    /// Css class to apply to the icon
11    /// </summary>
12    public string IconClass { get; set; }
13  }
```

2. The initialization module

The initialization module is invoked after the EPiServer.Shell.UIDescriptorRegistry is populated, and for each content type that have our custom attribute, append the css class to the IConClass attribute. This is, as per usual for attributes, done using reflection, but as it only happen once during site startup, the preformance impact is of no issue.

```
 1  /// <summary>
 2  /// Used in conjunction with <see cref="ContentIconAttribute"/> to ad
 3  /// </summary>
 4  /// <remarks><see cref="ContentIcon"/> and <see cref="ContentIconAttr
 5  [InitializableModule]
 6  [ModuleDependency(typeof(EPiServer.Cms.Shell.InitializableModule))]
 7  public class ContentTypeUIDescriptorInitializer : IInitializableModul
 8  {
 9   public void Initialize(InitializationEngine context)
10   {
11    var registry = context.Locate.Advanced.GetInstance<UIDescriptorRegi
12    var classes = GetDescriptorClasses();
13    foreach(var descriptor in registry.UIDescriptors) {
14     if(classes.ContainsKey(descriptor.ForType)) {
15      descriptor.IconClass += classes[descriptor.ForType];
16     }
17    }
18   }
19
20   public void Uninitialize(InitializationEngine context) {}
21
22   public void Preload(string[] parameters) {}
23
24   private Dictionary<Type, string> GetDescriptorClasses()
25   {
26    var types = AppDomain.CurrentDomain.GetAssemblies()
27         .SelectMany(ass => ass.GetTypes().Where(type => type.IsDefine
28
29    var descriptors = from type in types
30     select new {
31      type,
32      iconClass = ((ContentIconAttribute)Attribute.GetCustomAttribute(t
33     };
34
35    return descriptors.ToDictionary(key => key.type, value => value.ico
36   }
37  }
```

The initializer is pretty straightforward, it finds all classes with out ContentIcon attribute and use that to modify the registered UIDescriptors by appending our IconClass. One important detail is that it has a dependency on EPiServer.Cms.Shell.InitializableModule, which ensure it is initialized after the UIDescriptorRegistry is populated with all the UIDescriptors.

3. The icons

As all we can do is add a css class to each item in the page tree, we need icons that can be displayed using only a css class. This can be achieved in a number of ways, like adding the css for each icon, like so:

```
1  .myPageTypeIcon {
2    background: url('../Images/Icons/pageIcons16x16.png') 0px -16px no-r
3    height: 16px;
4    width: 16px;
5  }
```

This approach is almost as tedious as making one UI descriptor class for each page type, this is no good. A faster way is to use a icon library, where all icons are already configured. Here, we picked Icomoon (https://icomoon.io/), they have a free library of about 400 icons in a convenient woff font format.

I did make a few small changes to the css supplied, I added a prefix as the class names were very generic, and it didn't display entirely correct, the icons were compacted to 13px instead of 16px. Anyways, here is the base css for the icomoon font icons:

```
 1  [class^="icomoon-icon-"], [class*=" icomoon-icon-"] {
 2    font: normal normal normal 16px/1 icomoon;
 3    speak: none;
 4    text-transform: none;
 5    line-height: 1;
 6    /* Better Font Rendering =========== */
 7    -webkit-font-smoothing: antialiased;
 8    -moz-osx-font-smoothing: grayscale;
 9  }
10
11  .icomoon-icon-home:before {
```

```
12      content: "\e600";
13    }
14    …
15    .icomoon-icon-IcoMoon:before {
16      content: "\e7c2";
17    }
```

As you can see, I prefixed each icon with icomoon, and also modified the font directive in the base class slightly.

Once you have your icons, place everything somewhere under /ClientResources, In this example I placed them under clientresources/styles/icomoon, and modify your module.config file in the web root:

```
1    <clientResources>
2      <add name="epi-cms.widgets.base" path="Styles/icomoon/icomoon.css" r
3    <clientResources>
```

3b. Convenience factor - icon names

As there are a lot of icons, you can add constants for all the icons by using some clever search and replace, like so:

```
1    public class ContentIcon
2    {
3      /// <summary>
4      /// IcoMoon free icons, see more here https://icomoon.io/#preview-fr
5      /// </summary>
6      public static class IcoMoon
7      {
8        public const string accessibility = "icomoon-icon-accessibility";
9        public const string address_book = "icomoon-icon-address-book";
10       public const string aid = "icomoon-icon-aid";
11       public const string airplane = "icomoon-icon-airplane";
12       public const string alarm = "icomoon-icon-alarm";
13       public const string alarm2 = "icomoon-icon-alarm2";
14       …
15       public const string youtube = "icomoon-icon-youtube";
16       public const string youtube2 = "icomoon-icon-youtube2";
17       public const string zoomin = "icomoon-icon-zoomin";
18       public const string zoomout = "icomoon-icon-zoomout";
19     }
20   }
```

Using it all

After gluing it all together, you can now in a simple way add page tree icons to your content using just one attribute per content item. The constant class help you avoid misspelling anything, and you can also use the icomoon website as quick reference for how each icon look:
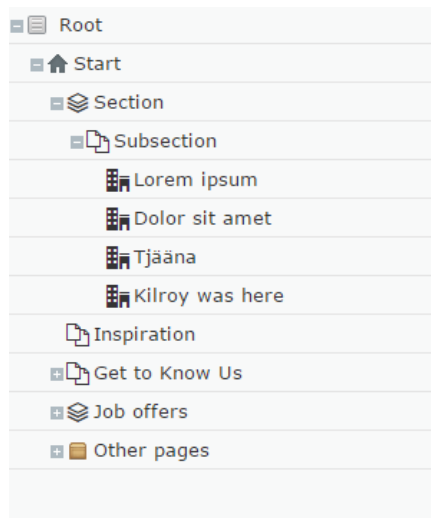
```
1    [ContentType(
2      GUID = "00000000-0000-0000-0000-000000000000",
3      DisplayName = "Section page",
4      GroupName = SystemTabNames.Settings)]
5    [ContentIcon(ContentIcon.IcoMoon.copy)]
6    public class MyVeryOwnPage : PageData
7    {
8    }
```

Finally

Here is the result in all its glory:

The fruits of our labor

Links

ThisIsNansen på Twitter

Nansen.se på svenska

Nansen.se in english

You can get the c# code source parts from here: http://pastebin.com/znYXReXc

Like   3 people like this. Sign Up to see what your friends like.

Tweet | 7

POSTED BY ERIK AT 18:26

LABELS: .NET , .NET 4.5 , C# , EPISERVER , EPISERVER 7.5 , EPISERVER CMS , ERIK NILSSON

2 comments

Add a comment

Top comments

Robert Linde 11 months ago - Shared publicly

Nice!

+1   1   · Reply

Kevin Candlert 7 months ago - Shared publicly

Awsome! :)

Although I had to narrow down the "GetAssemblies()" result. Or else it would crash on Castle DynamicProxies.

1   · Reply

Newer Post   Home   Older Post

Subscribe to: Post Comments (Atom)