



ALLAN THRAEN | 2 years ago | PDF |

[Tips and Tricks](#) [Optimizely \(Episerver\)](#) [CMS](#) [.NET Development](#)

OPTIMIZELY (EPISERVER): SPLIT FOLDER STRUCTURE FOR BLOCKS AND MEDIA



Since version 7 or 8 of Episerver (now Optimizely CMS), the shared Blocks and Media have been sharing the same folder structure. Some people see a benefit with the shared structure, and some absolutely hate it. Personally, I have gotten used to it - but I was recently asked if it's possible to split it up. Here's the hack I came up with.

First of all, fair warning: *This is very experimental and I'm not exactly sure of all the consequences of this hack, so if you plan to use it - do so at your own risk (but feel free to leave a comment below with the good, bad and ugly). It hasn't been tested in a multi-site scenario and most likely won't work ideal there (although the code should be pretty easy to adjust for that). Also, it doesn't migrate any existing blocks or media - so that should be done before hand.*

The basic approach I took was to create 2 folders in the root of the GlobalBlockFolder - one for Media and one for Blocks. Afterwards I basically just needed the Blocks and Assets widget to use the new folders as their root. They both use an `IContentRepositoryDescriptor` for that, so it was pretty straightforward to create a new descriptor for each (inheriting their original), overriding the root and then using an `IConfigurableModule` to intercept the original calls using Dependency Injection.

The example below is not very optimized - but it does seem to work. If the folders doesn't exist, it will create them - and then make sure the widgets use them.

```
1 using Episerver.Cms.Shell.UILUIDescriptors;
2 using Episerver.Framework;
3 using Episerver.Framework.Initialization;
4 using Episerver.ServiceLocation;
5 using Episerver.ServiceLocation.Compatibility;
6 using Episerver.Shell;
7 using System;
8 using System.Linq;
9
10 namespace ExperimentSite.Widgets
11 {
12     [InitializableModule]
13     [ModuleDependency(typeof(Episerver.Web.InitializationModule))]
14     public class Interceptornit : IConfigurableModule
15     {
16         public void ConfigureContainer(ServiceConfigurationContext context)
17         {
18             //Override / remove
19             context.Services.Intercept<IContentRepositoryDescriptor>(
20                 (locator, def) => (def is BlockRepositoryDescriptor) ?
21                     locator.GetInstance<MyBlockRepositoryDescriptor>() :
22                     (def is MediaRepositoryDescriptor) ? locator.GetInstance<MyMediaRepositoryDescriptor>() : def
23                 );
24         }
25     }
26
27     public void Initialize(InitializationEngine context)
28     {
29         //Add initialization logic, this method is called once after CMS has been initialized
30     }
31
32
33     public void Uninitialize(InitializationEngine context)
34     {
35         //Add uninitialization logic
36     }
37 }
38 }
```

Interceptornit.cs hosted with ❤ by GitHub

[view raw](#)

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Web;
5 using Episerver;
6 using Episerver.Cms.Shell.UILCompositeViews.Internal;
7 using Episerver.Cms.Shell.UILUIDescriptors;
8 using Episerver.Core;
9 using Episerver.Framework.Localization;
10 using Episerver.ServiceLocation;
11 using Episerver.Shell;
12 using Episerver.Web;
13
14 namespace ExperimentSite.Widgets
15 {
16     public class MyBlockRepositoryDescriptor : BlockRepositoryDescriptor
17     {
18         private const string NAME = "Blocks";
19         private readonly IContentRepository _repo;
20
21         private ContentReference _root;
22     }
23 }
```

```
24 public MyBlockRepositoryDescriptor(IContentRepository repo)
25 {
26     _repo = repo;
27
28     //Get or create root
29     ContentFolder rootfolder = _repo.GetChildren<ContentFolder>(ContentReference.GlobalBlockFolder).Where(f => f.Name == NAME).FirstOrDefault();
30     if (rootfolder == null)
31     {
32         rootfolder = _repo.GetDefault<ContentFolder>(ContentReference.GlobalBlockFolder);
33         rootfolder.Name = NAME;
34         _root = _repo.Save(rootfolder, EPiServer.DataAccess.SaveAction.Publish, EPiServer.Security.AccessLevel.NoAccess);
35     }
36     else _root = rootfolder.ContentLink;
37 }
38
39 public override IEnumerable<ContentReference> Roots
40 {
41     get
42     {
43         yield return _root;
44     }
45 }
46
47 }
48
49
50 public class MyMediaRepositoryDescriptor : MediaRepositoryDescriptor
51 {
52     private const string NAME = "Media";
53     private readonly IContentRepository _repo;
54     private ContentReference _root;
55
56     public MyMediaRepositoryDescriptor(IContentRepository repo)
57     {
58         _repo = repo;
59
60         //Get or create root
61         ContentFolder rootfolder = _repo.GetChildren<ContentFolder>(ContentReference.GlobalBlockFolder).Where(f => f.Name == NAME).FirstOrDefault();
62         if (rootfolder == null)
63         {
64             rootfolder = _repo.GetDefault<ContentFolder>(ContentReference.GlobalBlockFolder);
65             rootfolder.Name = NAME;
66             _root = _repo.Save(rootfolder, EPiServer.DataAccess.SaveAction.Publish, EPiServer.Security.AccessLevel.NoAccess);
67         }
68         else _root = rootfolder.ContentLink;
69     }
70     public override IEnumerable<ContentReference> Roots
71     {
72         get
73         {
74             yield return _root;
75         }
76     }
77
78 }
79 }
```

RepositoryDescriptors.cs hosted with ❤ by GitHub

view raw

Tips and Tricks

Optimizely (Episerver)

CMS

.NET Development

RECENT POSTS