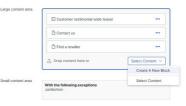
Tips and Tricks Optimizely (Episerver) CMS

AUTOMATICALLY HAVE OPTIMIZELY CONTENT CREATED IN THE CORRECT PLACE

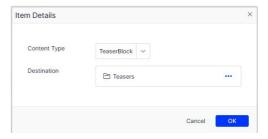


A classic problem is that you want better structure and governance in for example your blocks. Maybe you have a policy to have all Banners or FAQ items in a specific folder so they can easily be found and re-used across the site, but way too often editors take the 'easy' approach and use the "Create a New Block" option in a content area - which by default places the block in the "For this page" folder, rendering re-use impossible. Here is a simple solution to that.

My approach to solving the above problem is pretty straightforward - whenever a content is being created, simply check if it's being created in the right place - and if not, simply change the parent to make sure that it is.

This is easily done in the ContentEvents - and in the CreatingContent event you can modify the content before it's saved for the first time!

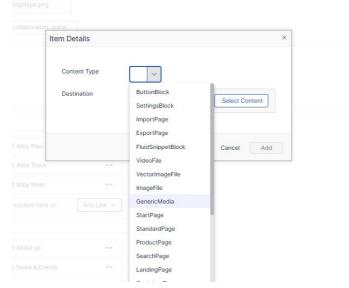
But - in order to make the solution a bit more generic and usable I've included a way to let editors define these rules on a settings page.

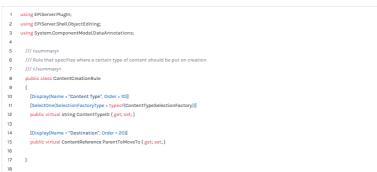


Note, this solution doesn't handle the fact that editors can move the content after creation - in which case it seems like a conscious decission. It simply just makes sure that the lazy 'create' button still places content where you want it.

I'd recommend putting the list of rules on a settings content item - in the below example it's on the SiteSettings tab of the start page - but that's for you to decide.

I also included a ContentTypeSelectionFactory. Note, there are many other ContentTypeSelectionFactories in other projects - but none included in the core of Optimizely CMS. It basically just produces a list of the content types for the property when you add a rule. A rule basically just that a certain type of content should go to a certain folder on creation.





```
19 [PropertyDefinitionTypePlugIn]
20 public class ContentCreationRuleList : PropertyList < ContentCreationRule > ()
 ContentCreationRule.cs hosted with ♥ by GitHub
       [ModuleDependency(typeof(InitializationModule))]
        public class ContentPlacerInit : IlnitializableModule
          public void Initialize(InitializationEngine context)
           cevents.CreatingContent += Cevents_CreatingContent;
            var rules = ServiceLocator.Current.Getinstance<!ContentLoader>().Get<:StartPage>(ContentReference.StartPage).ContentCreationRu
var rules=rules.FirstOrDefault(r > r.ContentTypeID == e.ContentContentTypeID.ToString(I));
            if (rule != null)
              if (!e.ContentParentLinkCompareTolgnoreWorkID(rule.ParentToMoveTo)) \\
                e.Content.ParentLink = rule.ParentToMoveTo;
 20
 23
24
 25
 ContentPlacerInit.cs hosted with ♥ by GitHub
  1 using EPiServer.Shell.ObjectEditing;
  3 public class ContentTypeSelectionFactory : ISelectionFactory
          public ContentTypeSelectionFactory(IContentTypeRepository repo)
         public | Enumerable < | Selectitem > GetSelections (ExtendedMetadata metadata)
            return _repo.List()
        .Select(r => new SelectItem() (Text = r.Name, Value = r.ID.ToString() ));
)
 ContentTypeSelectionFactory.cs hosted with ♥ by GitHub view raw
1 //_
3 [Display(GroupName = Globals.GroupNames.SiteSettings.)]
4 [EditorDescriptor(EditorDescriptor)
 5 public virtual IList<ContentCreationRule> ContentCreationRules ( get; set; )
6 7 //_
StartPage.cs hosted with ♥ by GitHub
Tips and Tricks Optimizely (Episerver) CMS
```

RECENT POSTS

CodeArt ApS

Teknikerbyen 5, 2830 Virum, Denmark Email: info@codeart.dk Phone: +45 26 13 66 96 CVR: 39680688

in O

Copyright © 2024