ALLAN THRAEN | 6 years ago | PDF |

C# Optimizely (Episerver) CMS Tips and Tricks

DESIGN PATTERN: TAG PAGES INSTEAD OF CATEGORIES

Episerver categories is one way to deal with taxonomy on a web site. But often I find that I prefer a simpler, more transparent approach of having Tag Pages replace them. Here's how.

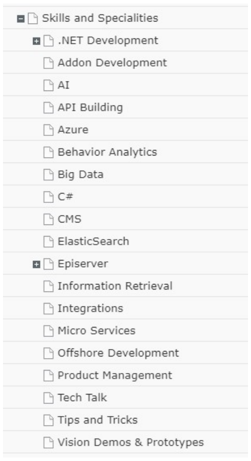
Episerver categories were nice. Back when they were introduced people were using Nokia 3310 (as pictured above), anyone who owned a house was a millionaire and we hadn't yet fully acknowledged how fast we are destroying the planet with global warming. You might even say, it was a better, simpler time. But things evolve and demands rise. And today I rarely think of using the old categories.

First of all, they haven't really evolved in the same pace as the rest of the CMS. They still live in Admin mode. The only way of translating them is through XML language files and they are just not really user friendly. To list elements in a category you either have to rely on external search such as Find or Vulcan or go the dark route of FindPagesWithCriteria (the evil that must not be named).

The concept of Tag Pages is really simple:

Have a page type called Tag Page (or whatever you feel is appropiate).
This is an easy one - just create a new page type.

Make a place in your content structure for your tags or taxonomy or categories. This is what mine looks like:



Skills and Specialities
- .NET Development
- Addon Development
- AI
- API Building
- Azure
- Behavior Analytics
- Big Data
- C#
- CMS
- ElasticSearch
- Episerver
- Information Retrieval
- Integrations
- Micro Services
- Offshore Development
- Product Management
- Tech Talk
- Tips and Tricks
- Vision Demos & Prototypes

Work with your taxonomy through Tag Pages just as you would with categories - except that now they are content with all the benefits that comes with that. Like support for multiple languages, permissions, publish flows and versioning.

Have a content area on your pages restricted to only accepting that content type. Now you can assign content to a taxonomy simply by dragging and dropping.

```
public class BlogPostPage : SitePageBase
{
    [AllowedTypes(AllowedTypes =new Type[] { typeof(TagPage)})]
    public virtual ContentArea Tags { get; set; }
}
```

If you want to list content in a given taxonomy you can simply look it up using the Softlinks repository. This is how I do it in my generic List Block that I can use to both list on tag pages or show descendant blog posts on other page types.

```
        IEnumerable<BlogPostPage> baseList = null;
        if (_loader.Service.Get<PageData>(p) is TagPage)
        {
            //If current page is a tag page, then look for softlinks, otherwise descendents
            var _soft = ServiceLocator.Current.GetInstance<IContentSoftLinkRepository>();

            var links = _soft.Load(p, true);
            baseList = links.Select(l => _loader.Service.Get<PageData>(l.OwnerContentLink)).Where
        }
        else
        {
            var descendents = _loader.Service.GetDescendents(p);
            baseList = descendents.Select(s => _loader.Service.Get<PageData>(s)).Where(s => s is I
        }
        baseList = FilterForVisitor.Filter(baseList).Cast<BlogPostPage>();
```

Make sure you in your renderings iterate through the items in the list - and that you do a full meta refresh after changes to get it displayed properly.

```
@using EPiServer.Core
@using EPiServer.Web.Mvc.Html

@model AllanTech.Web.Models.ViewModels.PageViewModel<AllanTech.Web.Models.Pages.BlogPostPage>

@Html.FullRefreshPropertiesMetaData(new[] { "Tags","HeaderImage"})
<div class="container">
    <div class="row">
        <!-- Latest Posts -->
        <main class="post blog-post col-lg-8">
            <div class="container">
                <div class="post-single">
                    <div class="post-thumbnail"><img src="@Html.ResizeImageWithFallback(Model.CurrentP
                    <div class="post-details">
                        <div class="post-meta d-flex justify-content-between" @Html.EditAttributes(x =
                            <div class="category">
                                @if (Model.CurrentPage.Tags != null)
                                {
                                    foreach (var tag in Model.CurrentPage.Tags.Items.Select(cai => cai
                                    {
                                        @Html.PageLink(tag)
                                    }
                                }
                            </div>
                        </div>
...
```

C# Optimizely (Episerver) CMS Tips and Tricks

Joel Abrahamsson's old take on this

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