

TIPS AND TRICKS EPISERVER

Site Map Generation with Custom Filter

Allan Thraen | 8 Months Ago | PDF | 0 | 21 Shares | [f](#) [t](#) [p](#) [e](#) [i](#) [l](#)

There's a handful of great add-ons I use in almost every Episerver project - and several of them are from Geta. Here is a useful hint if you, like me, use the Geta SiteMap generator.

An easy way to have the classic sitemap.xml generated for you, on your Episerver site, is of course to use the de-facto standard: Geta's open source Sitemap generator.

<https://github.com/Geta/SEO.Sitemaps/tree/master/src/Geta.SEO.Sitemaps>

Even though it's been around for quite a while I continue to be amazed at how well engineered it is - and how you can solve pretty much every need with it.

Recently I was looking for a way to have a property on the individual pages manage the visibility in the sitemap - and sure enough - that's an out-of-the-box feature.

And today I needed to write a bit of code to dynamically decide which pages should appear in the sitemap, based on some rather complex business rules. At first I assumed that I would probably have to fork the project to add this functionality, but luckily, it turns out that it implements it's own version of IContentFilter, managed by Episervers dependency injection. The default implementation does a lot of great filtering, so I simply opted to inherit from it and then extend on it with custom filtering logic.

```
[ServiceConfiguration(typeof(IContentFilter))]
public class CustomSiteMapContentFilter : ContentFilter
{
    public override bool ShouldExcludeContent(IContent content)
    {
        bool rt= base.ShouldExcludeContent(content);
        if (!rt)
        {
            //Do custom logic
            rt = (bool) content.Name.ToLower().StartsWith("a");
        }
        return rt;
    }
}
```

Note, that to make the above example easily readable I replaced my custom logic with a simple example to remove content with names starting with "a". Useless, sure - but would be fun to see if it's caught in test (in which case you might be testing too much) :-)

In the above example it's registered for dependency injection using attributes - however since only 1 IContentfilter is applied, it's important that it's registered last - so I later moved it to an IConfigurable initialization module set to run after the Sitemap had initialized.

TIPS AND TRICKS

EPISERVER



Previous Post
Publicwww - searching for interesting Episerver CMS use patterns

Next Post
The tricky redirect



Post Comments (0)

You might also like

About Allan

I'm a freelance software architect / developer and Episerver expert (EMVP) based in Copenhagen Denmark. I've worked for Episerver for more than 10 years before going freelance and understand both your business needs as well as the technical possibilities.



I love solving real business problems with innovative and creative coding!

Learn more about my skills and specialities, reach out to me at info@codeart.dk or use this form to contact me.



Blog posts

2020

March

February

January

2019

December

November

October

September

August

July

June

May

April

March

February

January

2018

December

November

October

September

August

June

Available Topics

.NET DEVELOPMENT

ADDON DEVELOPMENT

AI

API BUILDING

AZURE

BEHAVIOR ANALYTICS

BIG DATA

C#

CMS

CONTENTFUL

DIGIZUITE

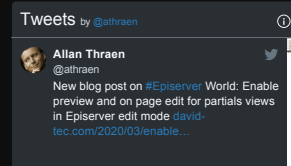
- ELASTICSEARCH
- EPISERVER
- FREELANCING
- INFORMATION RETRIEVAL
- INTEGRATIONS
- LIFE AS A CODER
- MICRO SERVICES
- OFFSHORE DEVELOPMENT
- PRODUCT MANAGEMENT
- TECH TALK
- TIPS AND TRICKS
- VISION DEMOS & PROTOTYPES

CodeArt ApS

Frederiksdalsvej 198, 2830 Virum, Denmark

Phone: +45 26 13 66 96

CVR: 39680688



Tweets by athraen