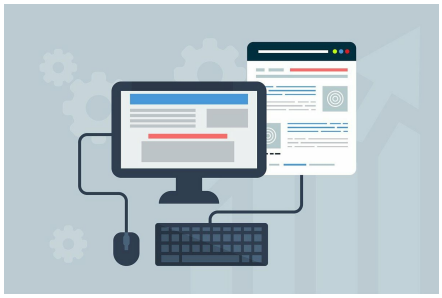
ALLAN THRAEN | 1 years ago | PDF |

Optimizely (Episerver) C# .NET Development Tips and Tricks

OPTIMIZELY CMS: LIST CONTENT RECURSIVELY ON A PAGE - AND LIST THE VISITOR GROUPS USED



Quite often a lot of the content on pages in Optimizely Content cloud (aka Episerver CMS) is structured in blocks placed in content areas. And often even blocks in blocks. Sometimes it's needed to quickly get a list of all the content items on a page - and sometimes you might also be interested in which visitor groups are used. Here is a couple of extension methods to help you with that.

Below are two extension methods that have proven themselfes quite useful to me when working with Optimizely/Episerver solutions.

The first will for a given IContent return all IContent used - including itself.

This means all blocks, media and so on that is in either ContentAreas or XhtmlStrings in it. Note, you might have other contentreferences - but they are not used here, so you'd have to extend it with logic for that.

An alternative approach is to use the IContentRepository.GetReferencesToContent(...) but that returns quite a lot more than what you often need. Same goes for the SoftLinks repository which has a slightly similar functionality.

The other extension method extends the IVisitorGroupRepository and can help you extract a list of visitor groups used in an IContent - namely in XhtmlStrings and ContentAreas which are the two places personalization is often applied. Note: personalization can of course also be used for content access control - that is not included in this method.

You can of course combine them - and that can be quite useful. For example if you want to know which visitor groups are used on a page (and it's blocks):

```
IContent c = _contentRepo.Get<IContent>(currentContent);
var referencedContent = _contentRepo.FetchReferencedContentRecursively(c);
var groups = referencedContent.SelectMany(rc => _vgRepo.ExtractVisitorGroups(rc)).Distinct().ToList();
```

Some usecases I have recently used them on includes:

- Generate reports with an overview of which visitor groups are used on which pages
- List suggestions for previewing a piece of content with the visitor groups that will take effect.
- Content collection for AI processing for an entire page.

Let me know what you can use them for - and watch out for future blog posts with more details on the usecases above.

```
using EPiServer;
using EPiServer.Core;
using EPiServer.Personalization.VisitorGroups;
using System;
using System.Collections.Generic;
using System.Linq;

namespace CodeArt.Optimizely.Extensions
{
    public static class UsefulExtensions.cs
    {
        ///<summary>
        /// Lists all content (typically blocks) used in content areas and xhtmlstrings on a content item, recursively (blocks in blocks, etc)
        ///</summary>
        public static IEnumerable<IContent> FetchReferencedContentRecursively(this IContentRepository repo, IContent c)
        {
            yield return c;
            foreach (var p in c.Property)
            {
                if (p.Value == null) continue;
                if (p.PropertyValueType == typeof(ContentArea))
                {
                    var ca = p.Value as ContentArea;
                    if (ca == null) continue;
                    foreach (var f in ca.Items)
                    {
                        foreach (var y in repo.FetchReferencedContentRecursively(repo.Get<IContent>(f.ContentLink)))
                            yield return y;
                    }
                }
                else if (p.PropertyValueType == typeof(XhtmlString))
                {
                    var ca = p.Value as XhtmlString;
                    if (ca == null) continue;
                    foreach (var f in ca.Fragments.Where(fr => fr is EPiServer.Core.Html.StringParsing.ContentFragment))
                    {
                        var j = f as EPiServer.Core.Html.StringParsing.ContentFragment;
                        foreach (var y in repo.FetchReferencedContentRecursively(repo.Get<IContent>(j.ContentLink)))
                            yield return y;
                    }
```

```
            }
        }

    }


    /// <summary>
    /// Analyzes which visitor groups are used in an IContent and returns a list of those groups.
    /// </summary>
    public static IEnumerable<VisitorGroup> ExtractVisitorGroups(this IVisitorGroupRepository vgRepo,IContent c)
    {
        foreach (var p in c.Property)
        {
            if (p.Value == null) continue;
            if (p.PropertyValueType == typeof(ContentArea))
            {
                var ca = p.Value as ContentArea;
                if (ca == null) continue;
                foreach (var f in ca.Items.Where(I => I.AllowedRoles != null && I.AllowedRoles.Any()))
                {
                    //Match! This page uses the visitor groups in I.AllowedRoles. Record.
                    foreach (var r in f.AllowedRoles)
                    {
                        yield return vgRepo.Load(Guid.Parse(r));
                    }
                }
            }
            else if (p.PropertyValueType == typeof(XhtmlString))
            {
                var ca = p.Value as XhtmlString;
                if (ca == null) continue;
                foreach (var f in ca.Fragments.Where(fr => fr is EPiServer.Core.Html.StringParsing.PersonalizedContentFragment))
                {

                    var j = f as EPiServer.Core.Html.StringParsing.PersonalizedContentFragment;
                    var roles = j.GetRoles();
                    foreach (var r in roles)
                    {
                        yield return vgRepo.Load(Guid.Parse(r));
                    }
                }
            }
        }


    }
}
```


Optimizely (Episerver) C# .NET Development Tips and Tricks

RECENT POSTS

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