ALLAN THRAEN | 1 years ago | PDF |

Tech Talk Tips and Tricks .NET Development Optimizely (Episerver) CMS

# CHRISTMAS COUNTDOWN: #11 DEPENDENCY INJECTION IS NOT AS EASY AS IT SEEMS


**Dependency Injection is an extremely useful pattern. It has been used with EPiServer CMS for years - and with .NET Core it has truly become the go-to method of coupling your business logic together. However, once you start having services depend on other services their lifetimes can give some unexpected difficulties.**

## The 2023 Christmas Countdown: 12 Common Pitfalls in Optimizely CMS - and how to avoid them

> *This blogpost is part of the 2023 Christmas Countdown series where I each day for the last 12 days before Christmas go through my Top 12 list of the most common and dangerous pitfalls I typically see in Optimizely (EPiServer) CMS 12 Implementations. If you want to learn more and perhaps have your own site evaluated feel free to reach out to us here at CodeArt.*
> Here are links to all the posts in this series as they are published:
> 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1

Todays count-down item is a throwback to a little bit of danger in a place you might not expect it.

> *Isolate business logic to services, register them in Service Container as Singleton, Scoped or Transient and use them through constructor injection. (Common 'best practice' approach)*

Now, most developers knows that an object registered as a singleton has a lifetime that follows the application, one registered as Scoped - typically has a lifetime that follows a predefined scope (in our web world it typically defaults to the request scope) and Transients are create new every time you request them. So far so good.

But - what happens when a Singleton service uses a Transient or Scoped service? Well - if they are used through constructor injection or property injection, they will essentially keep the value of when they were first instantiated by the Singleton - essentially making them singletons as well. Only, when they are fetched in methods as needed through the ServiceLocator (an approach that by many is often considered bad practice) will they actually follow their set scope.

Now - I realize this is a bit abstract, so let me explain why this is potentially a problem:

Let's say you have a service that loads and keeps data about the currently logged in user - perhaps on a commerce or self-service site - data that you use to show them their purchase history or other sensitive information you have stored about them - and you have of course set that as Scoped to keep it within the authorized request. That's all fine.

But - you also have another service - perhaps one that merges the personal data with the content on the site - and for performance reasons that is a Singleton. Now, if you are not very careful, you risk that it will show the wrong personal information to visitors.

And - to make matters worse, singletons are not just the singletons you know about and register. If - for example you have a custom Visitor Group Criteria that you use to personalize on those personal data (a very typicaly scenario), then keep in mind that without you know it, each criteria is actually a singleton behind the scenes - so if your scoped service is using constructor injection you are in for some wonderful and fun errors - that can be pretty hard to find and troubleshoot.

Read more here:
https://www.codeart.dk/blog/2023/1/when-best-practice-isnt-the-best---dependency-injection-and-optimizely-cms/

Tech Talk Tips and Tricks .NET Development Optimizely (Episerver) CMS

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