



ALLAN THRAEN |



5 months ago |



PDF |



Tech Talk Tips and Tricks CMS Website Improvements Optimizely (EpiServer)

CHRISTMAS COUNTDOWN: COMMON OPTIMIZEZY CMS PITFALLS - #12 PICKING THE RIGHT ADD-ONS



12 days to Christmas and here is my countdown list of the top 12 common pitfalls I see in Optimizely CMS implementations - along with some tips on how to avoid them. Today we'll take a look at #12 on the list: Picking the right add-ons

It's December once again, and as we approach the holidays I figured it was about time to get back to doing some blogging. Before summer I did a tech talk on the most common pitfalls I typically see in Optimizely (EpiServer) CMS implementations - and how to avoid them. Over the years I have gotten to see quite a lot of implementations and there are certain traps I notice clients fall into again and again. And now I figured - there's about 12 days until Christmas (we celebrate on the 24th here in Denmark) and perhaps a Top 12 Common pitfalls would make a nice Christmas countdown. That way we can all remember them and try to avoid them when we well rested and very full on holiday food move into 2024.

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



No. 12 Picking the right add-ons

On nuget.optimizely.com there are currently more than 1400 packages with add-ons. They exist in more than 19.000 versions and there's also some packages and add-ons on the regular nuget.org feed. Not to mention all the code samples, gists and other open source projects that haven't made it all the way to a package yet.

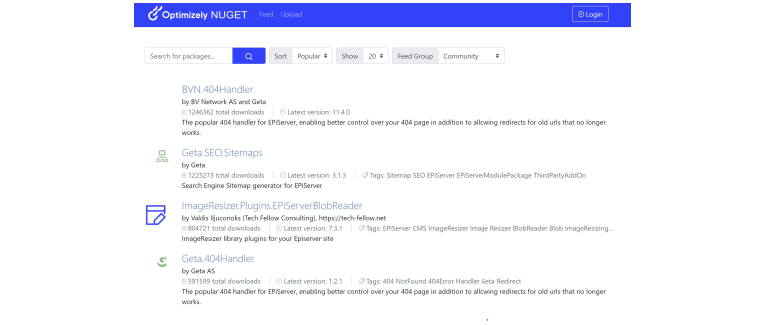
It's not wonder that sometimes people are struggling with knowing which add-ons to install - and every once in a while you pick one that is outdated, depend on something that can't be upgraded or is just not the right one for the job.

With experience you learn which add-ons are actively maintained and have become the defacto industry standards. Stuff like:

- BVN/GETA 404 Handler (not called NotFoundHandler)
- Geta Sitemaps
- DBLocalizationProvider
- Powerslice
- Geta ContentTypelcons
- Advanced reviews
- Most add-ons that are published by Optimizely / Are in the EpiServer namespace. Like EpiServer.Forms

But there are of course many other great add-ons - as well as many other add-ons that could have been

great, but never really made it. An approach I often use is to check out the most popular verified add-ons and the most popular community add-ons.



You can even drill into an add-on in that view and see how many releases they have had recently as well as which dependencies there are. I generally look for a nice steady and recent flow of releases that shows it's actively being maintained and EPiServer.CMS dependencies that shows which version of the CMS it's meant for. But of course it's not hard and fast rules - few releases recently could also just mean that it has reached a level of maturity.

On the other hand there are also a bunch of red flags that makes me typically consider if this add-on is really production ready:

- Clearly outdated (very old dependencies, old releases) - like Social Reach.
- Rarely used. Now, again - this could also just be because it's a very new add-on or hasn't gotten the buzz it deserved - but it's worth some more looking into
- 'Dead' add-ons - last updated years ago
- 'High impact' or 'Experimental' add-ons. It's fine to play around with these in your development environment, but be sure you know what you are doing if you use them in production.
- Add-ons that installs editorial routes and views outside of /EPiServer/ (or whatever your edit-mode path is called)
- Proprietary add-ons from a single partner that are not in a public nuget feed. They might be good, but there's probably a better, more standardized option that is not intended to tie you down to one implementation partner.

A final note is that while installing add-ons is a great way to enhance your CMS - you should also remember to maintain the site and remove add-ons that are not used. I often see a lot of add-ons installed that editors didn't even know was there - and might not need. Just installed because the developers figured 'why not'. Make sure they solve an actual problem, and that the users that have the problem (often the editors) know that the add-on can solve their problems.

Tech Talk | Tips and Tricks | CMS | Website Improvements | Optimizely (EpiServer)

RECENT POSTS