TYPO3 Core - Bug #64272

extbase query cache does not factor in start/stop times for pages

2015-01-14 12:46 - Nils Blattner

Status:	Closed	**Start date:**	2015-01-14
Priority:	Should have	**Due date:**	
Assignee:		**% Done:**	0%
Category:	Extbase	**Estimated time:**	0.00 hour
Target version:			
TYPO3 Version:	6.2	**Complexity:**	hard
PHP Version:	5.4	**Is Regression:**	No
Tags:		**Sprint Focus:**	

Description

Hi there

When extbase caches queries, it does not factor in start/stop times of pages that should be searched.

=> tx_myext_domain_model_something.pid IN (1285,1284,1283,1282,1281,1280)

If now page 1285 has a stop time in the future, it will not be removed and if there is a page 1289 with a start time in the future, it will not be added.
This can of course be mitigated with a lower time to live, however a different solution is desiarable. (Btw. the default lifetime is way too high)

=> How about the query cache collects start/stop times of the underlying pages (and possible pages) and factors them into the lifetime of the query cache entry
=> Additionally tags, that get deleted upon changes in the backend for these pages, is desirable

What do you guys think?

Can be reproduced:
=> Extbase Plugin with a page in the plugin options as data storage + recursive set
=> Subpage of the data storage which has a "start time" in the future and some entries inside
=> Call to the frontend => entries from said page are not shown (query cache entry is created now)
=> Wait until start time elapses
=> Clear frontend cache
=> Call to the frontend => entries are still not shown, even tough they should be

Cheers
Nils

History

#1 - 2015-01-14 14:09 - Mathias Schreiber

- Complexity set to hard

#2 - 2015-01-15 07:05 - Georg Ringer

This is not a bug, as this has never been checked, so nothing to do with the extbase query cache.

furthermore, there is no reason to check the start/stop of sysfolders (which is the place to store records). and there is the start/stop of records if you need that. it would be easy to have a scheduler task which copies the start/stop of pages into its record.

I am for closing this one.

#3 - 2015-01-19 15:48 - Nils Blattner

Hi Georg

Just because it has never been checked does not mean it is not a bug. And it has everything to do with the extbase query cache, because the process of getting items works fine without the query cache, but breaks if it is active. Also fe_groups access on a page level is most probably not checked properly because of it.

Imagine a sys folder with a certain fe_group access. This creates a race condition: clear cache > user with this specific access loads page > query cache now holds all the pages for everybody + the pages for his specific fe_groups combination which unpriviledged user can see afterwards.

It should be possible to activate/deactivate pages based on time/access and expect them work just like they advertise to an unwary editor.

Also the default query cache lifetime should be something realistic and not several years.

Best regards
Nils

#4 - 2017-06-13 11:40 - Benni Mack

- Status changed from New to Needs Feedback

The query cache was removed with v8 because of conceptual issues (and its migration to Doctrine), and it's simply not necessary anymore. Can you check if this helps?

#5 - 2017-09-14 11:10 - Riccardo De Contardi

- Status changed from Needs Feedback to Closed

90 days without feedback -> closing it.

If you think that this is the wrong decision or experience the issue again or have more information about how to reproduce it, please reopen it or open a new issue with a reference to this one. Thank you.