

TYPO3 Core - Feature #17630

gr_list concept needs to be improved

2007-09-27 10:45 - Michael Stucki

Status:	New	Start date:	2007-09-27
Priority:	Must have	Due date:	
Assignee:		% Done:	0%
Category:	Caching	Estimated time:	0.00 hour
Target version:		Complexity:	
PHP Version:	4.3	Sprint Focus:	Needs Decision
Tags:			
Description			
TYPO3 uses the "gr_list" which indicates the groups of which a frontend user is a member of.			
This information is used to identify cache records, indexed records, and probably more.			
If user A is a member of group 1 while user B is a member of groups 1 AND 2, both can't use the cached page which has been built for the other gr_list, even if their content is exactly the same.			
One of the resulting problems of this are:			
- page needs to be built multiple times (which is not needed in many situations)			
- users cannot find indexed content before someone else having the same gr_list has visited (=> indexed) the page (issue imported from #M6403)			
Related issues:			
Related to TYPO3 Core - Feature #17638: Integrate possibility to show content...		Closed	2007-09-29
Related to TYPO3 Core - Feature #17253: indexed_search should always include...		Closed	2007-04-27
Related to TYPO3 Core - Bug #16597: FE- Users with more than one group get wr...		Closed	2006-09-26
Related to TYPO3 Core - Bug #15250: t3lib_BEfunc::blindUserNames() does not w...		Closed	2005-11-30

History

#1 - 2007-09-27 10:46 - Michael Stucki

I have no idea how to solve this yet, so I'm just brainstorming for now...

#2 - 2007-09-27 10:55 - Michael Stucki

Idea [#1](#): Use "loginMode" flag to disable login information for a whole page.

Drawbacks: No personalized content is possible at all!

Considerations:

- I've made a patch already which can restore the login information for INT objects which are never cached anyways. However, this requires loading the full template (which is luckily stored in the cache, so it should not need to be re-parsed again).

#3 - 2007-09-27 11:10 - Michael Stucki

Idea 2: Introduce new flag which indicates if a page needs personalization, authorization (on page level), or none of both.

The flag has a default value and can be changed by extensions (similar to `$TSFE->set_no_cache()`).

If none of both is needed, then the gr_list of the page in the cache is set to "0,-3" (todo: maybe "0" is enough?) which means: No page access check is needed.

If only authorization is needed, then the gr_list of the page in the cache is set to "0,-4" (or "0,-3", see above) which means "lookup page access again".

If personalization is needed, the old behaviour will be used ("0,-1" applies to no logged in users, "0,-2,<groups>" applies to any logged in users matchin all specified groups).

Drawbacks:

- complex
- not sure how good it will work

#4 - 2007-09-27 12:45 - Steffen Kamper

Hi Michael,

this is really a problem that e.g. for properuser access it has to be done with USER_INT to prevent this cached information. I don't have a real idea atm how to solve.

It would help if \$TSFE->fe_user never be cached, but this is a complex thing. May be it's better to make this only instead of whole page method.

There were such a lot discussions about that and it's difficult to extract the essence out of it. Hope that a solution can be found.

#5 - 2007-09-28 16:27 - Ernesto Baschny

We struggle with the same problems on several customer sites. On most we have chosen "custom" search solutions over indexed_search exactly because of this limitation.

Some ideas also occurred in our team to improve that (specifically for indexed_search, but maybe it also can be used generally). Basically it would mean that the admin has to tell the indexed/cacher more about its page tree structure and which type of fe_group usage he has. For example:

- Be able as an admin to specify which fe_groups are relevant for a certain page tree. So saying in a certain page tree: only consider group "-1" (anonymous) and "2" (fullaccess), the cacher/indexed wouldn't care if the user is also in group 1, 3, 4 or whatever. We will always have maximum 2 cached entries and 2 indexed-search variants for any phash on that tree.

- Maybe it can help if we could say (in TypoScript for a specific page tree again): We don't use "user-specific content elements" in this page tree. So please just cache and index pages as a whole if the user has access to the page at all.

What we also usually require is a "fulltext search" that is able to scan the complete fulltext database without regarding gr_list at all. If the admin likes and configures it to be so the search form in "indexed search" will always find every content on the site (and never show "different" results for different user groups). When following the link will then either don't show the content or redirect to a login page instead or whatever. Some more complex configuration on that matter could "fine-tune" that behaviour (like a list of fe_group uid's to always include in the fulltext search results).

Also about the gr_list problematic there must be an easy way to "crawl" the whole site with different permissions settings. Is there? Using "crawler" I couldn't find an "easy" way to do it.

Well, so far for the "brainstorming"... :) Maybe some of the ideas here are not implementable at all, maybe they are.

#6 - 2007-11-20 14:03 - Valery Romanchev

this is related to #0005510 and #0004286

#7 - 2010-09-15 14:17 - Jonas Felix

Is there somewhere a documentation what 0,-2,-1 etc. means?

#8 - 2010-09-30 12:21 - Michael Stucki

@Jonas: 0,-1 means logged out. 0,-2 means logged in. The rest of that string contains the list of the usergroups.

#9 - 2013-05-05 21:34 - Alexander Opitz

- *Category deleted (Communication)*
- *Status changed from New to Needs Feedback*
- *Target version deleted (0)*

As this report is very old, is the handling in newer TYPO3 CMS Versions (like 6.0/6.1) more like you expect it?

#10 - 2013-09-06 09:20 - Alexander Opitz

- *Status changed from Needs Feedback to Closed*

No feedback for over 90 days.

#11 - 2013-09-06 09:45 - Michael Stucki

- *Status changed from Closed to New*

Hey Alexander,

the problem still exists, and I really like to get back on it one day. It would also be a nice GSOC project, for example. Therefore, please leave it open for the time being. It would still be great to have this one day...

Greetings, Michael

#12 - 2013-11-05 17:23 - Alexander Opitz

My idea to solve this problem:

- we save for which user groups this view is used instead of which user groups the user had who view this.
- this may collide with plugins which depends on the yet existing handling of the caching.

Examples of how it should work, to see if this is manageable:

On write

```
Page Access + User      => Access List
No FE Groups           => 0
    -1 + 0, -1         => -1
    -2 + 0, -2         => -2
    2 + 0, -2, 2       => 2
    2, 3 + 0, -2, 2, 3 => 2, 3
```

On Read

```
Access List + User      => read/no access
    0 + No FE Groups    => read
    -1 + 0, -1         => read
    -1 + 0, -2         => no access
    -2 + 0, -2         => read
    -2 + 0, -2, 2, 3   => read
    2 + 0, -2, 2       => read
    2, 3 + 0, -2, 2    => no access
    2, 3 + 0, -2, 2, 3 => read
```

A page can now have different content elements with different user groups. Situation, one page have three content elements:

Element A: Show for logged out users (-1)

Element B: Show for logged in users (-2)

Element C: Show for users with fe_group 2

Element D: Show for users with fe_group 3

```
User      => positive Access List + negative Access List
0, -1     => -1                      +
0, -2     => -2                      + 2, 3
0, -2, 2   => 2                      + 3
0, -2, 2, 3 => 2, 3                  +
0, -2, 3   => 3                      + 2
```

We need the negative list, cause for user "0, -2, 2, 3" we don't know if the cache entry with the positive Access List "2" is all he can get or do we need to generate a new page.

A page with content elements which have multiple user groups:

Element A: Show for users with fe_group 2 or 3

Element B: Show for users with fe_group 3 or 4

```
User      => positive Access List + negative Access List
0, -1     => 0                      + 2, 3, 4
0, -2     => 0                      + 2, 3, 4
0, -2, 2   => 2                      + 3, 4
0, -2, 2, 3 => 3                    +
0, -2, 3   => 3                    +
0, -2, 4   => 4                    + 2, 3
0, -2, 2, 4 => 3                    +
```

=> So we get a maximum of 4 pages in Cache.

#13 - 2015-01-21 09:43 - Mathias Schreiber

- Target version set to 8 LTS

#14 - 2016-07-11 16:55 - taywa gmbh

I would like to see an improvement in the gr_list stuff. If you have many different usergroups and combinations, its renders indexed_search useless, because you have to index for every combination. and this is exponential. thanks

#15 - 2017-02-23 14:50 - Riccardo De Contardi

- Target version changed from 8 LTS to 9.0

#16 - 2017-10-29 16:27 - Susanne Moog

- Category set to Image Cropping

- Sprint Focus set to PRC

#17 - 2017-10-29 21:47 - Michael Stucki

@Susanne Category "Image Cropping" seems like a mistake to me. I would rather say that this is about caching...

#18 - 2017-10-29 21:48 - Michael Stucki

@Susanne see my comment above (just added you as watcher to this issue).

#19 - 2017-10-30 08:10 - Susanne Moog

- *Category changed from Image Cropping to Caching*

#20 - 2017-10-30 08:11 - Susanne Moog

Yes, Image Cropping was a mistake, thanks for paying attention :)

#21 - 2018-01-28 14:29 - Susanne Moog

- *Target version deleted (9.0)*

#22 - 2020-03-24 17:52 - Benni Mack

- *Sprint Focus changed from PRC to Needs Decision*

#23 - 2020-10-08 17:13 - Benni Mack

OK. This is my proposal:

- We should cache on "CE level" or "per entity" level. Because if a CE does not have any restrictions (fe_group), it should be "ALWAYS CACHED". This should be cached away, and fetched.
- When "0,-1" is called, it *should* be possible to fetch everything statically already and just put the pieces in.
- If a page itself has pages.fe_group set, we need to differentiate between "Is allowed" or "is not allowed".

Basically we work like a "firewall" for the cache. Yes, cached under these circumstances. CEs / Pages should be based on the content, not on the visitor!