TYPO3 Core - Bug #85284

Workspace shows unchecked sectionIndex in preview

2018-06-15 20:41 - Patrick Gaumond

Status:	New	**Start date:**	2018-06-15
Priority:	Should have	**Due date:**	
Assignee:		**% Done:**	0%
Category:	Workspaces	**Estimated time:**	0.00 hour
Target version:			
TYPO3 Version:	8	**Complexity:**	nightmare
PHP Version:		**Is Regression:**	
Tags:		**Sprint Focus:**	

Description

If you uncheck sectionIndex (Show in Section Menus) of a tt_content it will still be shown in the Workspace preview.

It looks like the WS preview doesn't care about sectionIndex=0 and show it anyway.

Also true for the Menu sections in WS Preview: The tt_content will be in the list even with sectionIndex=0

Related issues:

Related to TYPO3 Core - Bug #85934: DatabaseQueryProcessor - doesn't respect ...	**Needs Feedback**	**2018-08-23**
Related to TYPO3 Core - Bug #88071: Language overlay in DatabaseQueryProcessor	**New**	**2019-04-04**
Related to TYPO3 Core - Bug #93336: Enabling translated content in workspace ...	**Closed**	**2021-01-21**

History

#1 - 2019-03-28 23:04 - Riccardo De Contardi

I tried the following test on version 10.0.0-dev (latest master)

Prerequisites

- TYPO3 with a "draft" workspace
- a page called "Page A"
- a page called "Page B"

Execution

1. in LIVE workspace, go to "Page A" and create a new element, CType: "section index (menu_section)"
   1. on the field "selected pages" > select "Page B", save and close
2. On "page B" create three content elements (CType: header)
   1. element 1 > [header] > write "Test 1 this should be visible in menu"
   2. element 2 > [header] > write "Test 2 this should **NOT** be visible in menu"
      1. on the tab "Appearance" > edit "Show in Section Menus[sectionIndex]" and put it to off
   3. element 3 > [header] > write "Test 3 this should be visible in menu"
3. Save and close
4. Go to "draft" workspace, go to "Page A" and preview it

Results

1. the link to the element "Test 2 this should **NOT** be visible in menu" is **NOT** visible > OK

Execution (test 2)

1. go to "Test 3 this should be visible in menu" and edit it
   1. on the tab "Appearance" > edit "Show in Section Menus[sectionIndex]" and put it to off
2. Save and close
3. Go to "draft" workspace, go to "Page A" and preview it

Results

1. the link to the element "Test 3 this should be visible in menu" is **STILL** visible > KO

#2 - 2020-07-30 14:54 - Christian Kuhn

Great reproduce information again, Riccardo. Thank you! Issue still confirmed in current master.

#3 - 2020-07-30 17:46 - Christian Kuhn

- Related to Bug #85934: DatabaseQueryProcessor - doesn't respect the translation config added

#4 - 2020-07-30 19:40 - Christian Kuhn

- Related to Bug #88071: Language overlay in DatabaseQueryProcessor added

#5 - 2020-07-30 20:18 - Christian Kuhn

Ok, here is the thing:

Content elements in core are rendered via fluid_styled_content (by default, since which version exactly?). In case of content element based sectionIndex menues, the according file is ext:fluid_styled_content/Configuration/TypoScript/ContentElement/MenuSection.typoscript

This file uses that snipped for the query:

```
...
                20 = TYPO3\CMS\Frontend\DataProcessing\DatabaseQueryProcessor
                20 {
                    table = tt_content
                    pidInList.field = uid
                    as = content
                    where = sectionIndex = 1
                    orderBy = sorting
                    dataProcessing {
                        10 = TYPO3\CMS\Frontend\DataProcessing\FilesProcessor
                        10 {
                            references.fieldName = image
                        }
                    }
                }
...
```

This triggers class DatabaseQueryProcessor with the query config. The class calls ContentObjectRenderer->getRecords() which creates a query via exec_getQuery() and receives the (always live, no language overlay, no workspace overlay) records. Those records are then overlayed with workspace versions and languages.

So the result rows DO have the sectionIndex=0 field set correctly, because they have been overlayed.

However, that row information is now too late and rows are NOT filtered again, eg. by given where restriction ... because, well, the query is done already. Thus, the record ("element 3" in Riccardos example) is rendered.

One could imagine a hack in ContentObjectRenderer->getQuery() trying to parse typoscript 'where' a second time to filter rows on PHP level. That approach however would be very limited:

- First only "additional restrictions" can be applied: Filtering rows that fall out of the query, but should still not be shows. If I turn the sectionIndex from above around and set sectionIndex=0 in live record but =1 in workspace, the approach of additional filtering would fail because the row wouldn't fall out of the query in the first place.
- Furthermore, a solution would require a parsing of a database where condition into some php pendant. This would be do-able for simple stuff like "fieldname = 42" or "fieldname > 42", but 'where' can be as complex as you want and also has stdWrap properties which could lead to everything ... In short, one does not want to create an sql parser in php ...
- The approach is also broken for stuff like LIMIT restrictions: "give me only x number of rows" - if a php side post-filter would kick rows, there may not be enough rows coming from the query to fill up with new candidates.

BTW: The old school menu_section index content object from AbstractMenuContentObject->sectionIndex() probably handled that case in a better way, since it considers the sectionIndex=1 restriction **after** overlays to language and version have been done ...

In general, the DatabaseQueryProcessor approach is very limited when it comes to language/version overlays and typically leads to wrong results. Issue #88071 is another example, and that one is not workspace but "just" language related. I'd actually expect more issues in this area to be reported in the future (please link those issues).

A good solution could be to do the overlays within the very same query, making the database doing overlays transparently (probably by using CTE's) in the first place. That approach would have other significant benefits, and we're thinking about that for some time already. It however requires deeper refactoring. Core v10 already has some important pre-work in this area to solve it, but the real work has not been done.

Thus, at the moment, this issue can't be solved in a good way since you'd always run into further not met scenarios. At the moment, I can only give the advise to somehow hack around those cases if they really need to be solved (for instance in this case by maybe falling back to the old school way of rendering). It would be great if according workarounds could be documented for others here if anyone comes up with something.

#6 - 2020-07-30 20:23 - Christian Kuhn

- Complexity set to nightmare

#7 - 2021-01-21 09:45 - Benni Mack

- Related to Bug #93336: Enabling translated content in workspace is not previewable added