

## TYPO3 Core - Bug #91697

### Access protecting root page results in "page not found" when user is not logged in, as opposed to "page inaccessible"

2020-06-23 13:01 - Claus Due

<b>Status:</b>	New	<b>Start date:</b>	2020-06-23
<b>Priority:</b>	Must have	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	0%
<b>Category:</b>	Frontend	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>		<b>Complexity:</b>	
<b>TYPO3 Version:</b>	8	<b>Is Regression:</b>	
<b>PHP Version:</b>		<b>Sprint Focus:</b>	
<b>Tags:</b>			
<b>Description</b>			
<p>This is a rather complex bug. It triggers when:</p> <ul style="list-style-type: none"><li>• You configure the root page of a domain to be access protected</li><li>• TypoScriptFrontendController attempts to resolve the page, but does not disable access checks</li><li>• An early guard clause triggers pageNotFound handling because page is not resolved - but root line is NOT empty</li><li>• This prevents the true group access checks from triggering</li></ul> <p>The result is that any custom class registered for page-not-found handling is told the page does not exist, when in fact it does exist but is merely access protected. And this in turn prevents such a custom class from handling this particular case as an access failure case.</p> <p>It is possible to work around this:</p> <pre>if (empty(\$parameters['pageAccessFailureReasons']) &amp;&amp; count(\$controller-&gt;rootLine) === 1) {     // No reason was provided and the root line contains a single entry. Check the only page that     is in the     // root line and determine if it was access protected. If it is access protected, return true.     // Perform the access check by repeating a call to get the page record, but with access check     enabled.     \$pageIsRootAndNotAccessibleByUserGroup = !\$controller-&gt;checkPageGroupAccess(\$controller-&gt;rootL     ine[0]); }</pre> <p>Suggested fix:</p> <p>In \TYPO3\CMS\Frontend\Controller\TypoScriptFrontendController::getPageAndRootline, in the second check for "if (empty(\$this-&gt;page))", check if:</p> <ul style="list-style-type: none"><li>• Root line contains a single entry (the requested page was the root page and it exists)</li><li>• \$this-&gt;checkPageGroupAccess(\$this-&gt;rootline<sup>0</sup>) returns false</li></ul> <p>If fulfilled:</p> <ul style="list-style-type: none"><li>• Set \$this-&gt;pageNotFound = 1 indicating that failure reason is that page is not accessible</li><li>• Set \$this-&gt;pageAccessFailureHistory['direct_access'][] = \$this-&gt;rootline<sup>0</sup>;</li><li>• call \$this-&gt;pageNotFoundAndExit('The requested page was not accessible!', \$GLOBALS['TYPO3_CONF_VARS']['FE']['pageUnavailable_handling_statheader']); instead of calling \$this-&gt;pageNotFoundAndExit('The requested page does not exist!');</li></ul> <p>This causes the same check and failure reason for a request made directly to a protected root page, as would trigger for a request made directly to a protected sub-page.</p> <p>Bug is encountered on 8.7 but almost certainly exists on all versions up to and after 10 LTS.</p> <p>Set to "must have" as this bug is a serious inconsistency that prevents proper use of custom page-not-available handling.</p>			

## History

---

#1 - 2020-06-23 13:06 - Claus Due

- Category set to Frontend