

Core - Bug # 53962

Epic # 55070 (Accepted): Workpackages

Epic # 55065 (New): WP: Overall System Performance (Backend and Frontend)

Bug # 52949 (Resolved): Speed decrease since 4.5

Status:	Resolved	Priority:	Should have
Author:	Alexander Stehlik	Category:	Performance
Created:	2013-11-25	Assigned To:	
Updated:	2014-02-14	Due date:	
TYPO3 Version:	6.2		
PHP Version:	5.4		
Complexity:			
Is Regression:	No		
Sprint Focus:			
Subject:	Class loader does not cache non existing classes		
Description	<p>Just found this one while testing https://review.typo3.org/25673/:</p> <pre>ValidatorResolver->buildBaseValidatorConjunction()</pre> <p>called by</p> <pre>ActionController->initializeActionMethodValidators()</pre> <p>will always trigger the class loader because it calls:</p> <pre>class_exists('string')</pre> <p>which of course is not a class and does not exist.</p> <p>Therefore it will not be stored in the cache and the whole class loader process starts all over again.</p> <p>This is done by every request to the page module (by the SysNote hook).</p> <p>Maybe it would make sense to store information about non existing classes in the cache?</p>		
Related issues:			
related to Core - Bug # 53970: getBaseValidatorConjunction is called for simp...		New	2013-11-26
related to Core - Task # 56086: Impexp functional tests do not fail on wrong ...		Resolved	2014-02-18

Associated revisions

Revision 67c4204d - 2014-02-14 16:34 - Alexander Stehlik

[TASK] Cache non existing classes

To prevent multiple retries when loading information about non existing classes (e.g. caused by class_exists() calls) the cache is filled with an empty string when a non existing class is detected.

With this information the class loader can return early and does not need to run all expensive checks for determining the class file.

Releases: 6.2

Resolves: #53962

Change-Id: I3c7e750bcce8846aaee79ac2d04527be2e31fc16

Reviewed-on: <https://review.typo3.org/25679>

Reviewed-by: Helmut Hummel

Tested-by: Helmut Hummel

History

#1 - 2013-11-25 23:43 - Markus Klein

- Parent task set to #52949

#2 - 2013-11-26 00:07 - Gerrit Code Review

- Status changed from New to Under Review

Patch set 1 for branch **master** of project **Packages/TYPO3.CMS** has been pushed to the review server.

It is available at <https://review.typo3.org/25679>

#3 - 2013-11-26 03:21 - Philipp Gampe

We should of course cache negative requests, but this sounds more like a bug in extbase?!?

#4 - 2013-11-26 10:57 - Alexander Stehlik

I guess you could consider this as a bug. I opened #53970 for this :)

#5 - 2014-01-20 08:38 - Thomas Maroschik

I am strongly against the caching of non existent classes. By doing so you break the class loader chain, if there are more than one class loader present.

Rather `class_exists` calls to simple types should be first checked via the `TypeHandlingUtility`.

#6 - 2014-01-20 09:54 - Alexander Stehlik

Which class loader chain do you mean?

Do you mean the native PHP chain when multiple autoloaders are registered? If so I can not see a problem because if the TYPO3 class loader fails the next one in the chain should take over, right?

Or is there a class loading chain inside the TYPO3 class loader? If so it should be possible to cache non existing classes after the chain.

I agree that the `TypeHandlingUtility` needs to be fixed. And I can not really think of a use case when you would run a lot of `class_exists` calls of non existing classes. But this does not mean it will not happen.

Do you see other problems besides the loader chain?

#7 - 2014-02-13 14:51 - Gerrit Code Review

Patch set 2 for branch **master** of project **Packages/TYPO3.CMS** has been pushed to the review server.

It is available at <https://review.typo3.org/25679>

#8 - 2014-02-14 16:33 - Gerrit Code Review

Patch set 3 for branch **master** of project **Packages/TYPO3.CMS** has been pushed to the review server.

It is available at <https://review.typo3.org/25679>

#9 - 2014-02-14 17:30 - Alexander Stehlik

- *Status changed from Under Review to Resolved*

- *% Done changed from 0 to 100*

Applied in changeset commit:67c4204dcb7131e1ed2b6aa02c2abdec2aa6ce93.