TYPO3 Core - Bug #75811

Backend speed decrease since 4.7

2016-04-20 14:03 - Christian Toffolo

Status:	Closed		**Start date:**	2016-04-20
Priority:	Must have		**Due date:**	
Assignee:			**% Done:**	0%
Category:	Performance		**Estimated time:**	0.00 hour
Target version:				
TYPO3 Version:	7		**Complexity:**	
PHP Version:			**Is Regression:**	No
Tags:			**Sprint Focus:**	

Description

I had the opportunity to return to work with the backend of TYPO3 4.7 after a long time working with 6.2 and 7.6

I felt the 4.7 noticeably more responsive and faster while the 7.6 seemed very heavy to navigate (load times, scroll speed) compared to 4.7. So I did some tests to try to understand why 7.6 is so slow.

Using the Task Manager of Chrome on the same frame of the backend in list module with 77 tt_address records:
4.7: Ram ~20M
7.6: Ram ~75M

Using the ispector on the frames showing the 77 records:
4.7: 19 requests / 405KB transfered / DOMContentLoaded 1,69s / Load: 2,13s
7.6: 59 requests / 285KB transfered / DOMContentLoaded 2,82s / Load: 4,28s

Counting the DOM elements of the frames with document.getElementsByTagName('*').length:
4.7: 2558
7.6: 4255

It looks like that 4.7 is 200% more performing, and it also felt like that. From what little I re-used 4.7, working on that was much more pleasant.

Related issues:

Related to TYPO3 Core - Bug #76581: Backend Performance Flaws	**New**	**2016-06-10**

History

#1 - 2016-04-20 16:45 - Markus Klein

I would say the number of DOM elements is simply owed to the necessary markup for Bootstrap for responsivness.

The number of requests is indeed interesting even though only 75% of data is transferred.

#2 - 2016-04-21 08:38 - Christian Toffolo

Markus Klein wrote:

> I would say the number of DOM elements is simply owed to the necessary markup for Bootstrap for responsivness.

Yes, I posted that data cause the more the DOM elements are the more the UI is laggy. Something to take into consideration designing a ui. Developers usually have powerful computers while editors usually have entry level ones or worst. The experience of them is different also because of this.

I just hope this bug report can help a little to improve the responsiveness of the BE in future versions or at least to not make the thing worst.

#3 - 2016-04-21 11:11 - Markus Klein

Well, seems you have some knowledge in that area.
Maybe wanna help with those issues?

#4 - 2016-04-21 11:21 - Christian Toffolo

Sure, I guess that I'm already helping with this bug report and I want to help more.

How can I help further?

#5 - 2016-04-21 11:40 - Markus Klein

If you can fix stuff with specific patches, just push them to our review system.
Also consider getting a Slack account for direct interaction in the typo3-coredev channel. Maybe also raise your findings there, since only a few people will actually stumble over this report here, but Slack has >2000 members.

#6 - 2016-04-27 14:11 - Frank Naegler

Hey,
thank you for the tests and numbers.
can you please provide some more information, especially:

- Operating System and version
- Browser(s) and Versions (can you test with some more browsers?)
- Internet connection, Provider, band width?
- hardware of your client: RAM? HDD/SSD? CPU?

It is interesting, because I had some performance tests yesterday in the office of a customer.
Same system, same WLAN, one chrome on windows, the other chrome on Mac OSX. The chrome on the mac was two times faster when reload the complete backend.

#7 - 2016-04-28 07:49 - Christian Toffolo

OS: Windows 10 (1511 10586.218)
Browser: Chrome 49.0.2623.112 m
Connection: ADSL (Low latency mode), NGI Italia, 7 Mb/s DL - 512 Kb/s
Hardware: 4GB RAM, 128GB SSD, Intel Core M 5Y10

Btw, this is a bug report related more to the responsiveness of the backend than the load time of the BE pages.
I think that we should focus on the resources (CPU/RAM) used to handle all that DOM+Javascript considering that editors usually have pretty outdated and slow machine.

#8 - 2017-04-03 20:56 - Christian Kuhn

- Status changed from New to Closed

i think we should close this issue: there is not that much we can do until various other parts of the backend are further resolved. with v8, extjs is only needed for the page tree, v9 will drop that. this should increase the rendering times significantly, along with a new tree implementation. only after that, we may have a detailed look if we can further improve the dom to see if that gains more speed. this should happen in dedicated tickets then.