

Browser caching problems

Michiel Roos posted an issue on Forge on December 12th 2014 with the subject “*pages SYS_LASTCHANGED does not update when page cache is cleared*”, which can be found at <https://forge.typo3.org/issues/63810>. Since we receive a lot of complaints at the Technical University of Eindhoven, which is related to browser caching, I've started to investigate this topic. The problem is a lot bigger than Michiel is mentioning. Alexander Opitz collected multiple issues in the parent task <https://forge.typo3.org/issues/63815>.

To start with: **TYPO3 has SERIOUS browser caching problems.**

In case of the university, and there must be much more websites, regular visitors will see no updates of pages for a very long time when the page itself does not change for a very long time. But, a page change is not only related to a change within the page itself, like content and page properties, but is depending on much more different factors. These can be:

- Changes in other pages, like titles for menus
- Changes in TypoScript
- Changes in HTML templates
- Changes in files, used by content elements
- and much more

All these changes are not registered by TYPO3, which means TYPO3 is telling the browser the page has not changed. When TYPO3 is doing that, the browser will serve the locally cached version of the page. This goes on as long as the page or records on that page do not change. For instance visitors will not see changes in menus and when you've changed the layout of the website by changing HTML templates, this will not be visible for these particular pages.

The problem

Mostly we want to have pages cached server side, but also client side. And with the last one we have a problem. When you want to have pages cached properly client side you use the setting “*config.sendCacheHeaders*”. If a page can be cached completely server side, it will send some headers to the browser as well:

- Last-Modified
- Date
- Expires
- ETag
- Cache-Control
- Pragma

The first one, “Last-Modified”, is the problem in our case. It has not been implemented well (understatement).

What does a browser do?

Read this very carefully because this part is important

- Visitor goes to page for the first time. Let's say this is version 1 of the page
- Headers are telling the browser the page can be cached, so browser stores it in local cache with the information send in the headers
- In the meantime the whole layout is changed of the website by using different HTML and TypoScript. Now we have a version 2 of this page
- When a visitor goes to the same page within the Expires time, the browser will not do a request, but simply serve the page (version 1) from the local cache.
- When a visitor goes to the same page after the Expires time, the browser will do a new request, with a request header "If-Modified-Since". This will contain the date which was set by the "Last-Modified" header with the very first request (version 1).
- TYPO3 does not detect a change in the page, since changes are only registered for page properties and records within this page, and tells the browser "Last-Modified" is still the same as with the initial request.
- The browser will get the page from local cache again (version 1) and show it to the visitor. (Although a newer version is send to the browser)
- Every time this visitor will go to that page, the browser will serve it from its own cache (version 1), until the page properties or a record within this page has been changed by an editor. The visitor will never see version 2.

The only way a visitor will notice a change is when he/she clears the browser cache, because the initial version with the "Last-Modified" header is deleted.

Where does it go wrong?

A mentioned before, the "Last-Modified" header needs to be changed whenever there is a change related to that page. Not only the page properties or records located on that page, but every change which should be reflected on that page, like menus, TypoScript or layout.

The field "**SYS_LASTCHANGED**" in the "pages" table is responsible for the date/time in the "Last-Modified" header. The field was introduced by Kasper and is in there from at least the initial revision he made on October 3rd 2003. You can discuss the naming of this field, but it was intentionally used for exactly the purpose of tracking page related changes. The Last-Modified header was introduced since December 20th 2004, so it is there for a very long time.

During rendering \TYPO3\CMS\Frontend\ContentObject\ContentObjectRenderer will check for each content element if the tstamp of this content element is bigger than the one of the page. If so, SYS_LASTCHANGED is updated for that particular page. That's done in \TYPO3\CMS\Frontend\ContentObject\ContentObjectRenderer::lastChanged().

Two cObjects are adapted to use the method lastChanged() as well, because these are content objects which are handling other content objects. These two are RECORDS and CONTENT.

But ...

There are way more cObjects and other topics which should reflect changes to that page, like:

- starttime and endtime of pages and content
- Fluid (File changes, in particular layouts and partials)
- All MENUs - (Depending on type of menu all pages/content within the menu should be checked if tstamp is higher than SYS_LASTCHANGED in the page showing the menu)
- FILE (File changes)
- FILES (File changes)
- FLUIDTEMPLATE (File changes)
- IMAGE (File changes)
- IMG_RESOURCE (File changes)
- IMGTEXT (File changes)
- MEDIA (File changes)
- MULTIMEDIA (File changes)
- QTOBJECT (File changes)
- SVG (File changes)
- SWFOBJECT (File changes)
- TEMPLATE (File changes)
- All file changes for external files like JS, CSS, HTML, TypoScript
- Changes up in rootline, like TypoScript changes

Where to go now?

The whole problem can be split in three:

- Changes related to cObjects
- Changes of everything outside, like config settings and external TypoScript files
- Changes related to extensions

The first one needs to be solved by doing checks on the changes which happen in cObjects.

The second one can be automated, or can be quickly solved by embedding the solution which Michiel is proposing. That means that administrators or integrators need to clear the cache when a system wide change has been made.

The third one needs documentation. We need to educate extension developers they should use the lastChanged() method when they are showing information which is not directly related to the page, like records from a storage page. In fact, we do not need TCEMAIN.clearCacheCmd! This can be automated with lastChanged().

The quickest solution would be to get rid of the “Last-Modified” header, but that is avoiding the problem, although this is only a lame cache validator. A short term solution is mentioned by Michiel Roos in <https://forge.typo3.org/issues/63798#note-1>.

Add the following to .htaccess:

```
<IfModule mod_headers.c>
    Header unset Last-Modified
</IfModule>
```

Not using this header will not give performance problems on a server. TYPO3 is rendering the whole page anyway when using this header, simply because it has to define if the header should change.

Another issue is that when the Last-Modified response header is equal to If-Modified-Since request header, it should return a “304 Not Modified” status without any content, instead of the “200 OK” it is sending now, including the content. But this is minor.

Why did we not notice?

We, as developer, administrator or integrator, are used to flush the cache of a browser by clearing the whole browser cache or by using a hard refresh in the browser on a page. Also when logged in in TYPO3, there is no cached version of a page. This way the changes will be reflected in our browsers.

We also tell our clients they might have a browser caching problem. We are so used to this and grew up with it. In fact the fault is not the browser, it is us.

A visitor will never, or at least almost never, clear the browser cache or do a hard refresh in their browser. They are not aware that a page is constantly served from their local browser cache and not refreshed with a newer version from TYPO3.

Conclusion (personal)

IMHO we should remove the “Last-Modified” header for now and start implementing it properly. Doing so will take some time and the issue is too big to ignore for a while. It will only be used when the TYPO3 caching time already has expired, so the page will be rendered again anyhow.

Just my two cents.

Cheers,

Patrick Broens