

## Redmine - Defect #29625

### application.css imported by themes not covered by cache control versioning

2018-09-17 17:47 - T Goeg

Status:	New	Start date:	
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:	Themes	Estimated time:	0.00 hour
Target version:		Affected version:	3.4.6
<b>Description</b>			

The subject and [#24617-12](#) pretty much says it all.

After an update, the old application.css still gets used by browsers' caches, with no means other than renaming the file and its import statements in the used theme to force cache invalidation.

Partial workaround:

public/themes/<activetheme>/stylesheets/application.css

Change import statement

@import url(../../../../stylesheets/application.css);

to some other name (e.g. application\_v2.css)

Rename public/stylesheets/application.css to e.g. application\_v2.css

Also apply the cache control hack in the mentioned ticket.

However, you can still never go back to application.css as browsers will still keep it cached forever. A fix is needed that makes the filename unique on every update, much like all other resources.

This may also affect other resources delivered by themes (e.g. images), so a best practice should be given for how to avoid that as well.

#### Related issues:

Related to Redmine - Defect #24617: Browser js/css cache remains after upgrade

Closed

#### History

##### #1 - 2018-09-18 01:28 - Go MAEDA

- Subject changed from application.css not covered by cache control versioning to application.css imported by themes not covered by cache control versioning

- Category set to Themes

##### #2 - 2018-09-18 01:28 - Go MAEDA

- Related to Defect #24617: Browser js/css cache remains after upgrade added

##### #3 - 2019-03-15 15:44 - Yuuki NARA

+1

I think this information should be known more.

##### #4 - 2021-11-05 21:48 - Jonathan Cormier

Change import statement

@import url(../../../../stylesheets/application.css);

to some other name (e.g. application\_v2.css)

Rename public/stylesheets/application.css to e.g. application\_v2.css

A "better" workaround is to append a ?version tag instead of renaming the file, that way you don't have to modify redmine internal files. The downside still being that you have to remember to update your theme everytime you upgrade redmine..

Example:

```
@import url(../../../../stylesheets/application.css?4.3.2);
```

So I also modified the cache directive from <https://www.redmine.org/projects/redmine/wiki/BrowserCaching#The-work-round-for-RedMine>

I decided to stop caching any files which don't have the ?version tags. At least on redmine 4.2.3, the vast majority of the files have this tag. Only leaving out a few images which are loaded from application.css

```
- # the regex logic: after either /javascripts/ or /stylesheets/ find the suffixes we want, followed by any quantity of numbers 0-9
+ # the regex logic: after either /javascripts/ or /stylesheets/ find the suffixes we want, followed by one or more numbers 0-9
  # This works because the files we want to cache always appear after one of those 2 directories: but not the files we want to ignore
  # /journals/edit/24174.js and /uploads.js?attachment_id=1&filename=my-file-to-upload.png
- location ~* {{REDMINE_RELATIVE_URL_ROOT}}(?<file>/(?:(:plugin_assets/|themes/).+)?(?:javascripts|stylesheets|images|favicon)/.+(?:css|js|jpe?g|gif|ico|png|html)(\?[0-9]+)?$) {
+ location ~* {{REDMINE_RELATIVE_URL_ROOT}}(?<file>/(?:(:plugin_assets/|themes/).+)?(?:javascripts|stylesheets|images|favicon)/.+(?:css|js|jpe?g|gif|ico|png|html)(\?[0-9]+)$) {
```