

Redmine - Defect #35872

Attendee listbox ordering problem solution

2021-09-13 10:58 - Osman Yozgatlioglu

Status:	New	Start date:	
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:	Issues	Estimated time:	0.00 hour
Target version:		Affected version:	4.1.4
Resolution:			
Description			
!!			

History

#1 - 2021-09-13 11:18 - Osman Yozgatlioglu

- File 35872-attendee-ordering-patch.diff added

Sorry for premature submit.

We're using Redmine 4.1.4 stable with Turkish locale. We faced with wrong ordering attendee list at issues as shown as

2021-09-13_11-52.png

I find patch #16730 and generate new one to solve our issue. Our ordering solved like;

2021-09-13_11-56.png

Basically changed some "sort" operations with "sort_by{|user|}"

You can find our patch file. attachment:35872-attendee-ordering-patch.diff

Install details;

Environment:

Redmine version 4.1.4.stable
Ruby version 2.6.7-p197 (2021-04-05) [x86_64-linux]
Rails version 5.2.6
Environment production
Database adapter PostgreSQL
Mailer queue ActiveJob::QueueAdapters::AsyncAdapter
Mailer delivery smtp

SCM:

Subversion 1.7.14
Git 1.8.3.1

#2 - 2021-09-14 22:29 - Holger Just

With your patch, you effectively disable sorting in Ruby. As such, this patch does not entirely solve the issue but introduces other subtle issues. In the two methods you have changed there, this will often results in many members of the list to actually be sorted correctly (by your database via `Principal.sorted`).

However, at least in `Issue#assignable_users`, the author and assignee would always be put at the end if they are added in addition to regular assignable users.

The current behavior of the sort of an array containing principals (i.e. users or groups) uses the custom `Principal#<=>` method. Here, we use `String#casecmp` to get an order of display names. Unfortunately, this method is not entirely usable with non-ASCII Unicode. To quote the docs here:

| *Currently, case-insensitivity only works on characters A-Z/a-z, not all of Unicode.*

A workaround which might get partly better results might be to use something like this in `Principal#<=>` instead:

```
self.to_s.downcase(:fold) <=> principal.to_s.downcase(:fold)
```

This uses Unicode case-folding to get a lower-case version of the strings to compare. Unfortunately, this still is not 100% correct for Turkish characters (i.e. the special handling of the dotless *i*). However, Ruby does not appear to support combining Unicode and Turkish case folding right now.

As such, it might be necessary to instead rely on the database to sort things correctly instead (if they can do a better job than Ruby here). This might have to be investigated at least for the "usual" databases MySQL, Postgres and SQLite, plus any other database currently supported by Redmine. If we want to go this route, we would have to adapt the list-building methods to only rely on the database for the entire sorting. Additional users (such as the author or current assignee) would thus have to be passed to the generated SQL query and can't be appended later.

Files

2021-09-13_11-56.png	28.1 KB	2021-09-13	Osman Yozgatlioglu
2021-09-13_11-52.png	25.9 KB	2021-09-13	Osman Yozgatlioglu
35872-attendee-ordering-patch.diff	1.11 KB	2021-09-13	Osman Yozgatlioglu