Redmine - Feature #11709

"Issues assigned to me" ordering priority

2012-08-26 19:05 - William Roush

Status: New Start date:

Priority: Normal Due date:

Assignee: % Done: 0%

Category: My page Estimated time: 0.00 hour

Target version:

Resolution:

Description

I see it as just being ordered by priority.

I think two other things should be taken into account:

- Issue's due date.
- Issue's target version due date.

I have issues that have either no target set, or a target set with no due date that are higher up than issues that are due next week due to the target version they're attached to.

It'll make this page MUCH more useful.

Related issues:

Related to Redmine - Feature #12223: Make "My Page"-Widgets filterable Closed

Related to Redmine - Feature #7769: Sortable columns in issue lists on "My page" Closed 2011-03-03

History

#1 - 2012-08-27 12:39 - Jean-Philippe Lang

Sorting by due date first would make sense indeed. I propose to sort by due date/version due date then priority, is it OK for you?

#2 - 2012-08-27 15:59 - William Roush

I agree with that, this may be also a view that we'd want available on the main issue list page, so it may be best to put this query where both the "my page" and "view issues" page can call it.

I was looking into writing it but my Ruby-fu is quite weak and this query is a bit messy. :(

#3 - 2012-12-14 16:38 - Tobias Fischer

Wouldn't it be nice if widgets have the same wellknown filter abbility like on the Tickets page? This is proposed in ticket #12223

#4 - 2013-06-27 14:42 - Michel Albert

I just wanted to have this feature implemented myself.

This is what I came up with...

NOTE: This will most likely only work on PostgreSQL!!!

2025-05-17 1/2

```
<% if assigned_issues.length > 0 %>
<%= link_to 1(:label_issue_view_all), :controller => 'issues',
```

#5 - 2013-06-28 09:22 - Daniel Felix

Well another idea is to implement user defined filters. With this way, the user could define his views by himself.

#6 - 2013-06-28 10:20 - Michel Albert

Here's a slightly better version. It sorts by "score" first, then by status and last update time. Essentially it only adds the "score" ordering.

```
diff -ur a/app/views/my/blocks/_issuesassignedtome.html.erb b/app/views/my/blocks/_issuesassignedtome.html.erb
--- a/app/views/my/blocks/_issuesassignedtome.html.erb 2013-06-27 14:36:39.111675538 +0200
+++ b/app/views/my/blocks/_issuesassignedtome.html.erb 2013-06-27 14:36:08.423136285 +0200
@@ -4,7 +4,7 @@
                            :conditions => {:assigned_to_id => ([User.current.id] + User.current.group_id
s)},
                                :limit => 10.
                                :include => [ :status, :project, :tracker, :priority ],
                                :order => "#{IssuePriority.table_name}.position DESC, #{Issue.table_name}.upd
ated_on DESC") %>
                                :order => "(#{IssuePriority.table_name}.position * EXTRACT(EPOCH FROM AGE(#{I
ssue.table_name}.due_date)))    DESC NULLS LAST, #{IssuePriority.table_name}.position DESC, #{Issue.table_name}.u
pdated_on DESC") %>
 <%= render :partial => 'issues/list_simple', :locals => { :issues => assigned_issues } %>
 <% if assigned_issues.length > 0 %>
<%= link_to l(:label_issue_view_all), :controller => 'issues',
```

#7 - 2020-04-12 10:21 - Go MAEDA

- Related to Feature #7769: Sortable columns in issue lists on "My page" added

2025-05-17 2/2