redmineorg-copy202205 - Vote #72411

"Issues assigned to me" ordering priority

2022/05/09 16:20 - Admin Redmine

:	New		:		2022/05/09
:			:		
:			:		0%
:	My page_42		:		0.00
:			:		0.00
Redmineorg_URL:	https://www.redmine.org/issues/11709		**status_id:**		1
category_id:	42		**tracker_id:**		2
version_id:	0		**plus1:**		0
issue_org_id:	11709		**affected_version:**		
author_id:	41332		**closed_on:**		
assigned_to_id:	0		**affected_version_id:**		
comments:	7				

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.

journals

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?

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. :(

Wouldn't it be nice if widgets have the same wellknown filter abbility like on the Tickets page?

This is proposed in ticket #12223

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!!!

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] + Use
                                                   :limit => 10,
                                                   :include => [:status, :project, :tracker, :priority],
-   :order => "#{IssuePriority.table_name}.position DESC, #{Issue.t
+   :order => "(#{IssuePriority.table_name}.position * EXTRACT(E

<%= link_to l(:label_issue_view_all), :controller => 'issues',

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

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
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] + Use
                                                         :limit => 10,
                                                         :include => [ :status, :project, :tracker, :priority ],
-                                                        :order => "#{IssuePriority.table_name}.position DESC, #{Issue.t
+                                                        :order => "(#{IssuePriority.table_name}.position * EXTRACT(E
ue_date))) DESC NULLS LAST, #{IssuePriority.table_name}.position DESC, #{Issue.table_name}.updated_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',

---

related_issues

relates,Closed,12223,Make "My Page"-Widgets filterable
relates,Closed,7769,Sortable columns in issue lists on "My page"

#1 - 2022/05/10 17:15 - Admin Redmine

- *My page_42*