

### "Issues assigned to me" ordering priority

<b>Status:</b>	New	<b>Start date:</b>	
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	0%
<b>Category:</b>	My page	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>			
<b>Resolution:</b>			

**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.

<b>Related issues:</b>		
Related to Redmine - Feature #12223: Make "My Page"-Widgets filterable	<b>Closed</b>	
Related to Redmine - Feature #7769: Sortable columns in issue lists on "My page"	<b>Closed</b>	<b>2011-03-03</b>

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

**#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 ability 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!!!

```
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") %>
<%= render :partial => 'issues/list_simple', :locals => { :issues => assigned_issues } %>
```

```
<% if assigned_issues.length > 0 %>
<p class="small"><%= 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_ids) },
+
+      :limit => 10,
+
+      :include => [ :status, :project, :tracker, :priority ],
+
+      :order => "#{IssuePriority.table_name}.position DESC, #{Issue.table_name}.updated_on DESC") %>
+
+      :order => "(#{IssuePriority.table_name}.position * EXTRACT(EPOCH FROM AGE(#{Issue.table_name}.due_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 %>
+
+      <p class="small"><%= 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