Redmine - Feature #23641

Increase the number of sort columns in custom queries

2016-08-23 10:52 - Anton Kalinin

Status:	New		**Start date:**	
Priority:	Normal		**Due date:**	
Assignee:			**% Done:**	0%
Category:	Issues filter		**Estimated time:**	0.00 hour
Target version:				
Resolution:				

Description

It would be great to be able to set more than 3 sort columns in custom queries. Maybe, 10 might be enough?

Here is a scenario to sort issues I'm going to use in my organization, it lacks the amount of sort columns:

1. Group issues by the priority field;
2. Sort by the Parent task field to visualize the issue hierarchy;
3. Sort by the Status field descending. The nearer an issue is to be done, the higher it should be in the list. It can help not to keep a lot of incomplete issues.
4. Sort by the % Done field descending due to the consideration above.
5. Sort by the # field ascending to order them in the creation order.

History

#1 - 2016-09-13 07:12 - Toshi MARUYAMA

- Description updated

#2 - 2016-09-27 10:41 - Andrey Lobanov (RedSoft)

Anton Kalinin wrote:

> It would be great to be able to set more than 3 sort columns in custom queries. Maybe, 10 might be enough?
>
> Here is a scenario to sort issues I'm going to use in my organization, it lacks the amount of sort columns:
>
> 1. Group issues by the priority field;
> 2. Sort by the Parent task field to visualize the issue hierarchy;
> 3. Sort by the Status field descending. The nearer an issue is to be done, the higher it should be in the list. It can help not to keep a lot of incomplete issues.
> 4. Sort by the % Done field descending due to the consideration above.
> 5. Sort by the # field ascending to order them in the creation order.

You can change it easily in your redmine copy.

In your app/models/query.rb (changed : .slice(0, 3) to .slice(0, arg.length))

```
def sort_criteria=(arg)
  c = []
  if arg.is_a?(Hash)
    arg = arg.keys.sort.collect { |k| arg[k] }
  end
  if arg
    c = arg.select { |k, o| !k.to_s.blank? }.slice(0, arg.length).collect { |k, o| [k.to_s, (o == 'desc' ||
o == false) ? 'desc' : 'asc'] }
  end
  write_attribute(:sort_criteria, c)
end
```

And in app/views/queries/_form.html.erb:
change 3 to any number

```
<fieldset><legend><%= l(:label_sort) %></legend>
```

```
<% 3.times do |i| %>
```

```
<% 3.times do |i| %>
```