

## Redmine - Feature #5010

### Hours per project per user

2010-03-08 16:52 - Jiří Křivánek

<b>Status:</b>	New	<b>Start date:</b>	2010-03-08
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	0%
<b>Category:</b>	Roadmap	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>			
<b>Resolution:</b>			
<b>Description</b>			
<p>I am strongly missing the estimated hours per project and per user. The attached image shows one of the pages of the Roadmap project tab. Quite perfect but it just displays the numbers of issues per user (completed slash total). I need (or better to say my management requires) the hours (completed slash total).</p> <p>Any chance to have this simple feature soon?</p>			

### History

#### #1 - 2010-03-08 17:23 - Jiří Křivánek

- File redmine-req-1.png added

Even better place for the time estimate is demonstrated by the second attached image. Well, I can realize that the previous was just easy but this one is a bit difficult...

#### #2 - 2010-03-08 22:53 - Jiří Křivánek

Perhaps this could help:

I updated the versions\_helper.rb file, the render\_issue\_status\_by method this way:

```
def render_issue_status_by(version, criteria)
  criteria ||= 'category'
  raise 'Unknown criteria' unless STATUS_BY_CRITERIAS.include?(criteria)

  h = Hash.new {|k,v| k[v] = [0, 0]}
  if criteria == 'assigned_to'
    begin
      # Total estimated hours
      Issue.sum("estimated_hours",
        :group => criteria,
        :conditions => ["#{Issue.table_name}.estimated_hours is not NULL AND #{Issue.table_name}.fixed_version_id = ?", version.id]).each {|c,s| h[c][0] = s.to_i}
      # Open issues count
      Issue.sum("(estimated_hours * (100 - done_ratio)) / 100",
        :group => criteria,
        :include => :status,
        :conditions => ["#{Issue.table_name}.estimated_hours is not NULL AND #{Issue.table_name}.fixed_version_id = ? AND #{IssueStatus.table_name}.is_closed = ?", version.id, false]).each {|c,s| h[c][1] = s.to_i}
    rescue ActiveRecord::RecordNotFound
      # When grouping by an association, Rails throws this exception if there's no result (bug)
    end
  else
    begin
      # Total issue count
      Issue.count(:group => criteria,
        :conditions => ["#{Issue.table_name}.fixed_version_id = ?", version.id]).each {|c,s| h[c][0] = s}
      # Open issues count
      Issue.count(:group => criteria,
        :include => :status,
        :conditions => ["#{Issue.table_name}.fixed_version_id = ? AND #{IssueStatus.table_name}.is_closed = ?", version.id, false]).each {|c,s| h[c][1] = s}
    rescue ActiveRecord::RecordNotFound
      # When grouping by an association, Rails throws this exception if there's no result (bug)
    end
  end
end
```

```
end
end
counts = h.keys.compact.sort.collect {|k| {:group => k, :total => h[k][0], :open => h[k][1], :closed => (h[k][0] - h[k][1])}}
max = counts.collect {|c| c[:total]}.max

render :partial => 'issue_counts', :locals => {:version => version, :criteria => criteria, :counts => counts, :max => max}
end
```

It appears to do what I needed: It displays the estimated hours rather than the issues counts.

Could you please include my piece of code with one of the next Redmine releases, as another option pickable from the dropdown?

Thank you & best regards,

Jiri.

### #3 - 2010-11-01 22:37 - Jiří Křivánek

Any progress about this?

### #4 - 2012-03-29 19:09 - Jiří Křivánek

Any progress about this?

## Files

---

redmine-req.png	191 KB	2010-03-08	Jiří Křivánek
redmine-req-1.png	266 KB	2010-03-08	Jiří Křivánek