

## ChiliProject - Bug # 306: Make the query list in the sidebar a ul

<b>Status:</b>	Declined	<b>Priority:</b>	Normal
<b>Author:</b>	Felix Schäfer	<b>Category:</b>	User interface
<b>Created:</b>	2011-03-25	<b>Assignee:</b>	Felix Schäfer
<b>Updated:</b>	2011-12-17	<b>Due date:</b>	
<b>Remote issue URL:</b>			
<b>Affected version:</b>			
<b>Description:</b>	The query list in the sidebar of the issue views currently is a bunch of links with @ @ in between, anyone against changing that to an unordered list styled to have no bullets?		

### History

#### 2011-03-25 04:41 pm - Felix Schäfer

Oh, the current code is at source:/app/views/issues/\_sidebar.rhtml#L17.

#### 2011-03-25 06:54 pm - Felix Schäfer

- Target version changed from 1.3.0 to 2.0.0

I realized this can't go into the 1.x line because it breaks how you'd need to use the view hooks in the issue sidebar. The proposed patch could be something like (I've inspected all sidebars to make sure the @#sidebar ul@ css rule wouldn't disturb anything elseâ€¦):

```
<pre><code class="diff">diff --git a/app/views/issues/_sidebar.rhtml b/app/views/issues/_sidebar.rhtml
index db85f97..946e556 100644
--- a/app/views/issues/_sidebar.rhtml
+++ b/app/views/issues/_sidebar.rhtml
@@ -1,23 +1,25 @@
<h3><%= l(:label_issue_plural) %></h3>
-<%= link_to l(:label_issue_view_all), { :controller => 'issues', :action => 'index', :project_id => @project, :set_filter => 1 } %><br />
+<ul>
+<li><%= link_to l(:label_issue_view_all), { :controller => 'issues', :action => 'index', :project_id => @project, :set_filter => 1 } %></li>
<% if @project %>
-<%= link_to l(:field_summary), :controller => 'reports', :action => 'issue_report', :id => @project %><br />
+<li><%= link_to l(:field_summary), :controller => 'reports', :action => 'issue_report', :id => @project %></li>
<% end %>
<%= call_hook(:view_issues_sidebar_issues_bottom) %>
-
<% if User.current.allowed_to?(:view_calendar, @project, :global => true) %>
- <%= link_to(l(:label_calendar), :controller => 'calendars', :action => 'show', :project_id => @project) %><br />
+<li><%= link_to(l(:label_calendar), :controller => 'calendars', :action => 'show', :project_id => @project) %></li>
<% end %>
<% if User.current.allowed_to?(:view_gantt, @project, :global => true) %>
- <%= link_to(l(:label_gantt), :controller => 'gantts', :action => 'show', :project_id => @project) %><br />
+<li><%= link_to(l(:label_gantt), :controller => 'gantts', :action => 'show', :project_id => @project) %></li>
<% end %>
<%= call_hook(:view_issues_sidebar_planning_bottom) %>
+<ul>

<% unless sidebar_queries.empty? -%>
<h3><%= l(:label_query_plural) %></h3>
-
+<ul>
<% sidebar_queries.each do |query| -%>
```

```

-<%= link_to(h(query.name), :controller => 'issues', :action => 'index', :project_id => @project, :query_id => query) %><br />
+<li><%= link_to(h(query.name), :controller => 'issues', :action => 'index', :project_id => @project, :query_id => query) %></li>
<% end -%>
+</ul>
<%= call_hook(:view_issues_sidebar_queries_bottom) %>
<% end -%>
diff --git a/public/stylesheets/application.css b/public/stylesheets/application.css
index 7e5847b..f60f9d2 100644
--- a/public/stylesheets/application.css
+++ b/public/stylesheets/application.css
@@ -75,6 +75,7 @@ h4, .wiki h3 {font-size: 13px;padding: 2px 10px 1px 0px;margin-bottom: 5px; bord
#sidebar hr{ width: 100%; margin: 0 auto; height: 1px; background: #ccc; border: 0; }
* html #sidebar hr{ width: 95%; position: relative; left: -6px; color: #ccc; }
#sidebar .contextual { margin-right: 1em; }
+#sidebar ul {list-style-type: none; padding: 0; margin: 0;}

#content { width: 75%; background-color: #fff; margin: 0px; border-right: 1px solid #ddd; padding: 6px 10px 10px 10px; z-index: 10; }
* html #content{ width: 75%; padding-left: 0; margin-top: 0px; padding: 6px 10px 10px 10px;}</code></pre>

```

This just doesn't look "nice" to me though. I propose switching those lists (and essentially every more or less dynamic sidebar list/menu) to RedmineMenus, that way the elements could be declared in the models/controllers they rely on (the calendar and gantt links for example) and it would be easier to extend them from plugins than with the ViewHooks. I also suspect (haven't tested it though) RedmineMenu to be a little more performant than the ViewHooks in this instance. Comments?

**2011-03-28 09:58 pm - Eric Davis**

Felix Schäfer:

The code looks good but you are correct, anything using the hooks will need to be changed.

> This just doesn't look "nice" to me though. I propose switching those lists (and essentially every more or less dynamic sidebar list/menu) to RedmineMenus, that way the elements could be declared in the models/controllers they rely on (the calendar and gantt links for example) and it would be easier to extend them from plugins than with the ViewHooks.

-1 for using RedmineMenus for this. The menus come with a lot of other code that wouldn't be used.

If there was a simpler ChiliProjectList or something that was an application-wide array, that would be better. Kinda like how the SCMs are registered but we'd need a key for where the List belongs too (e.g. @:view\_issues\_sidebar => [...], :view\_wiki\_sidebar => [...], :view\_footer => [...])@

> I also suspect (haven't tested it though) RedmineMenu to be a little more performant than the ViewHooks in this instance. Comments?

I dunno, that would need testing. Menu adds some mid sized objects with every request. ViewHooks has classes. The Menu would be easier to understand though...

**2011-05-29 06:18 pm - Eric Davis**

- Target version deleted (2.0.0)

Pushing it out of 2.0.0, at least for the RC. We could add it post-RC or in a minor release.

**2011-12-16 08:28 pm - Eric Davis**

Felix:

I think the saved queries should be submenu items (part of #559). If they are, then I don't think we'd need this change anymore. What do you think?

**2011-12-17 11:19 am - Felix Schäfer**

*- Status changed from Open to Declined*

Submenu sounds great :-)