

Redmine - Defect #33548

Column header is clickable even when the column is not actually sortable

2020-06-03 17:15 - Vincent Robert

Status:	Closed	Start date:	
Priority:	Normal	Due date:	
Assignee:	Go MAEDA	% Done:	0%
Category:	Issues list	Estimated time:	0.00 hour
Target version:	4.0.8	Affected version:	4.1.1
Resolution:	Fixed		
Description			
Please find below a patch to fix the 'sortable?' method in QueryColumn class.			
Currently, the method returns true if @sortable is False. Therefore, some headers display a link when the column is actually not sortable (multi-value custom-fields columns in Projects page for instance).			
Thank you for your work			
<pre>diff --git a/app/models/query.rb b/app/models/query.rb index 0dec7d211..c3311ffbf 100644 --- a/app/models/query.rb +++ b/app/models/query.rb @@ -51,7 +51,7 @@ class QueryColumn # Returns true if the column is sortable, otherwise false def sortable? - !@sortable.nil? + !@sortable.blank? end def sortable</pre>			

Associated revisions

Revision 20783 - 2021-03-14 04:14 - Go MAEDA

Fix that column header is clickable even when the column is not actually sortable (#33548).

Patch by Vincent Robert and Marius BALTEANU.

Revision 20784 - 2021-03-14 07:57 - Go MAEDA

Merged r20783 from trunk to 4.1-stable (#33548).

Revision 20785 - 2021-03-14 07:59 - Go MAEDA

Merged r20783 from trunk to 4.0-stable (#33548).

History

#1 - 2020-06-04 18:03 - Go MAEDA

- Category set to Issues list
- Status changed from New to Confirmed
- Target version set to Candidate for next minor release

#2 - 2020-06-05 04:40 - Go MAEDA

Thank you for the fix. I have confirmed the issue.

Here is a test to catch the issue.

```
diff --git a/test/unit/query_test.rb b/test/unit/query_test.rb
index e4da5e1aa..c068de162 100644
```

```

--- a/test/unit/query_test.rb
+++ b/test/unit/query_test.rb
@@ -1629,6 +1629,8 @@ class QueryTest < ActiveSupport::TestCase
     field.update_attribute :multiple, true

     q = IssueQuery.new
+   column = q.available_columns.detect {|c| c.name.to_s == 'cf_1'}
+   assert !column.sortable?
     assert !q.sortable_columns['cf_1']
   end

```

#3 - 2020-06-05 06:24 - Mischa The Evil

- *Description updated*

I haven't looked deeply into this, but this makes me wonder: did this method ever worked correctly? And if so: what changed and was it intentional?

#4 - 2020-06-05 08:53 - Go MAEDA

Mischa The Evil wrote:

I haven't looked deeply into this, but this makes me wonder: did this method ever worked correctly? And if so: what changed and was it intentional?

I still don't know the cause, but it appears that 3.3 or later are affected. In my test, 2.6 through 3.2 worked as expected.

#5 - 2020-06-07 08:30 - Mischa The Evil

- *Assignee set to Mischa The Evil*

I'll do some research to see if I can dig up some (historical) context for this issue.

Note: if it's already clear for someone what's exactly going on here and what the proper way to fix it would be, feel free to jump-in/take-over. I just want to make sure that we're fixing the cause of the issue at its root instead of quick-fixing it with a patch while it's not well-understood.

#6 - 2021-02-27 13:41 - Marius BĂLTEANU

- *Target version changed from Candidate for next minor release to 4.0.8*

I'm assigning this to [4.0.8](#), we should fix it.

#7 - 2021-02-27 13:52 - Marius BĂLTEANU

I took a look in the code, but I didn't find what changed the behaviour in [3.3.0](#).

Anyway, looking to the implementation, this line (<source:trunk/app/models/query.rb#L123>) never returns nil because of the `|| false`. To be more specific:

```
self.sortable = custom_field.order_statement || false
```

If `custom_field.order_statement` returns nil, then the expression returns false.

From my point of view, the patch proposed by Vincent is safe, but I've added it to the CI to check the test results.

Another fix that should work is the following

```

# Returns true if the column is sortable, otherwise false
def sortable?
-   !@sortable.blank?
+   @sortable
end

```

#8 - 2021-02-27 14:15 - Marius BĂLTEANU

The tests pass with both fixes:

1. <https://gitlab.com/redmine-org/redmine/-/pipelines/262768044>
2. <https://gitlab.com/redmine-org/redmine/-/pipelines/262768066>

#9 - 2021-03-08 10:10 - Markus Boremski

Should we change the Target-Version?

#10 - 2021-03-13 15:14 - Marius BĂLTEANU

- File 0003-Add-test-for-33548.patch added
- Assignee changed from Mischa The Evil to Go MAEDA

Mischa The Evil wrote:

I'll do some research to see if I can dig up some (historical) context for this issue.

Note: if it's already clear for someone what's exactly going on here and what the proper way to fix it would be, feel free to jump-in/take-over. I just want to make sure that we're fixing the cause of the issue at its root instead of quick-fixing it with a patch while it's not well-understood.

Mischa, please let us know when you find something.

Until then, I'm assigning this to Go Maeda in order to include the fix in [4.0.8](#). The patch proposed by Vincent is safe from my point of view, I'm attaching a test case. We can refactor then in a major version.

#11 - 2021-03-14 04:18 - Go MAEDA

- Subject changed from Fix column header when column is not sortable to Column header is clickable even when the column is not actually sortable
- Status changed from Confirmed to Resolved
- Resolution set to Fixed

Committed the patch. Thank you.

#12 - 2021-03-14 07:59 - Go MAEDA

- Status changed from Resolved to Closed

Files

patch.diff	326 Bytes	2020-06-03	Vincent Robert
0003-Add-test-for-33548.patch	945 Bytes	2021-03-13	Marius BĂLTEANU