

## Redmine - Defect #8371

### MySQL error when filtering a custom field using the REST api

2011-05-13 20:05 - Bruno Bigras

<b>Status:</b>	Closed	<b>Start date:</b>	2011-05-13
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	0%
<b>Category:</b>	REST API	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	1.3.0	<b>Affected version:</b>	
<b>Resolution:</b>	Fixed		
<b>Description</b> MySQL 5.1.41 Redmine 1.1.3 ruby 1.8.7 rails 2.3.5  I created a custom field named 'sha1' (text format), 0 to 40 length.  I tested it with a dumb value 'aaa' and it worked fine with <a href="http://myhost/redmine/issues.xml?project_id=5&amp;cf_1=aaa">http://myhost/redmine/issues.xml?project_id=5&amp;cf_1=aaa</a>  I had an error when I tried to use it with a real sha1 hash.  I was able to reproduce the problem when trying to filter with a value of 'c' but it worked with 'a' and 'b'. <a href="http://myhost/redmine/issues.xml?project_id=5&amp;cf_1=a">http://myhost/redmine/issues.xml?project_id=5&amp;cf_1=a</a> <-- works <a href="http://myhost/redmine/issues.xml?project_id=5&amp;cf_1=b">http://myhost/redmine/issues.xml?project_id=5&amp;cf_1=b</a> <-- works <a href="http://myhost/redmine/issues.xml?project_id=5&amp;cf_1=c">http://myhost/redmine/issues.xml?project_id=5&amp;cf_1=c</a> <-- doesn't works  I get from the log:  <pre>Processing IssuesController#index to xml (for 10.1.10.240 at 2011-05-13 13:43:19) [GET]   Parameters: {"format"=&gt;"xml", "project_id"=&gt;"5", "action"=&gt;"index", "controller"=&gt;"issues", "cf_1"=&gt;"c"}</pre> <pre>Query::StatementInvalid: Mysql::Error: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near ') AND projects.id = 5 AND projects.status=1 AND projects.id IN (SELECT em.projec' at line 1:</pre> There's no condition after the <b>where</b> at line 14. (note that I indented the query myself).  <pre>SELECT count(DISTINCT `issues`.id) AS count_all FROM `issues` LEFT OUTER JOIN `issue_statuses` ON `issue_statuses`.id = `issues`.status_id LEFT OUTER JOIN `projects` ON `projects`.id = `issues`.project_id WHERE   ((issue_statuses.is_closed=0) AND   issues.id IN (     SELECT issues.id     FROM issues     LEFT OUTER JOIN custom_values ON       custom_values.customized_type='Issue' AND       custom_values.customized_id=issues.id AND       custom_values.custom_field_id=1     WHERE   ) AND   projects.id = 5 AND   projects.status=1 AND   projects.id IN (SELECT em.project_id FROM enabled_modules em WHERE em.name='issue_tracking'))</pre>			

#### Associated revisions

Fixed shot filter expression parsing depending upon field operators (#8371).

Removed test part failing with PostgreSQL (#8371).

Restored valid test part removed with r7625 (#8371).

Filter out illegal values to prevent raise of PostgreSQL exceptions, restored last test part removed with r7625 (#8371).

**#1 - 2011-05-13 20:28 - Bruno Bigras**

Also note that the sha1 I was trying to use with the filter also began with the letter 'c'.

- File 2013 fix custom field filter.patch added

I figured out that the regex at [query.rb#L270](#) (line 3 here) removes the 'c' (and other letters) from the filter value.

```
def add_short_filter(field, expression)
  return unless expression
  params = expression.scan(/^(o|c|!|\*|!|\*)?(.*)$/).first
  add_filter field, (params[0] || "="), [params[1] || ""]
end
```

<http://www.redmine.org/projects/redmine/repository/entry/trunk/app/models/query.rb#L270>

Here's a diff that fixes my problem. I guess a better solution would be to only use the regex when it's needed.

```
Index: redmine-1.1.3/app/models/query.rb
=====
--- redmine-1.1.3.orig/app/models/query.rb      2011-04-29 05:33:42.000000000 -0400
+++ redmine-1.1.3/app/models/query.rb           2011-05-19 10:00:21.265646521 -0400
@@ -254,8 +254,13 @@

```

```
def add_short_filter(field, expression)
  return unless expression
  -   parms = expression.scan(/^(o|c|!|*|!|\\*)?(.*)$/).first
  -   add_filter field, (parms[0] || "="), [parms[1] || ""]
  +   if field =~ /^cf_(\d+)$/
  +     # custom field
  +     add_filter field, "=", [expression]
  +   else
  +     parms = expression.scan(/^(o|c|!|*|!|\\*)?(.*)$/).first
  +     add_filter field, (parms[0] || "="), [parms[1] || ""]
  +   end
end
```

- Target version set to Candidate for next minor release

Nice catch !

- File short filters *regexps.patch* added

Tried to write a more generic patch.

- File short filters *regexps.patch* added

Previous patch was incomplete.

#6 - 2011-05-20 15:23 - Etienne Massip

- File deleted (short\_filters\_regexprs.patch)

#7 - 2011-05-20 19:26 - Bruno Bigras

Etienne Massip wrote:

Previous patch was incomplete.

The patch seems to work perfectly. I'll report back if I have any problem later.

Thanks!

#8 - 2011-10-13 09:38 - Etienne Massip

- Target version changed from Candidate for next minor release to 1.3.0

BTW I'm not sure it's worth it to maintain such complexity in request arguments processing as there is another way to do so (via f[], o[], and v[] arguments), maybe should we stick to handling equality.

I'm Ok with reverting these changes if needed.

#9 - 2011-10-17 18:07 - Etienne Massip

- Status changed from New to Resolved

- Resolution set to Fixed

#10 - 2011-10-19 20:20 - Etienne Massip

- File filter\_out\_illegal\_query\_filter\_values.patch added

I reverted [r7628](#) because of failing tests in QueryTest unit test.

The tests failed because the values used in applied filters are not values that are displayed in the issue query form.

I attach the patch for history.

Anyway, the commit was not immediatly related to this issue, which remains Fixed.

#11 - 2011-11-13 10:24 - Jean-Philippe Lang

- Status changed from Resolved to Closed

Files

2013_fix_custom_field_filter.patch	770 Bytes	2011-05-19	Bruno Bigras
short_filters_regexprs.patch	4.21 KB	2011-05-20	Etienne Massip
filter_out_illegal_query_filter_values.patch	1.98 KB	2011-10-19	Etienne Massip