

Redmine - Defect #8371

MySQL error when filtering a custom field using the REST api

2011-05-13 20:05 - Bruno Bigras

Status:	Closed	Start date:	2011-05-13
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:	REST API	Estimated time:	0.00 hour
Target version:	1.3.0	Affected version:	
Resolution:	Fixed		
Description			
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 http://myhost/redmine/issues.xml?project_id=5&#38;cf_1=aaa			
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'. http://myhost/redmine/issues.xml?project_id=5&#38;cf_1=a <-- works http://myhost/redmine/issues.xml?project_id=5&#38;cf_1=b <-- works http://myhost/redmine/issues.xml?project_id=5&#38;cf_1=c <-- 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"=>"xml", "project_id"=>"5", "action"=>"index", "controller"=>"issues", "cf_1"=>"c"} 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 where 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</pre>			

```
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'))
```

Associated revisions

Revision 7624 - 2011-10-13 01:45 - Etienne Massip

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

Revision 7625 - 2011-10-13 08:18 - Etienne Massip

Removed test part failing with PostgreSQL (#8371).

Revision 7627 - 2011-10-13 08:42 - Etienne Massip

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

Revision 7628 - 2011-10-18 19:51 - Etienne Massip

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

History

#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'.

#2 - 2011-05-19 16:37 - Bruno Bigras

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

#3 - 2011-05-19 17:28 - Etienne Massip

- Target version set to Candidate for next minor release

Nice catch !

#4 - 2011-05-20 15:08 - Etienne Massip

- File *short_filters_regexp.patch* added

Tried to write a more generic patch.

#5 - 2011-05-20 15:23 - Etienne Massip

- File *short_filters_regexp.patch* added

Previous patch was incomplete.

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

- File *deleted (short_filters_regexp.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_regexp.patch	4.21 KB	2011-05-20	Etienne Massip
filter_out_illegal_query_filter_values.patch	1.98 KB	2011-10-19	Etienne Massip