

Redmine - Defect #24156

Redmine might create many AnonymousUser and AnonymousGroup entries

2016-10-24 17:31 - Holger Just

Status:	Closed	Start date:	
Priority:	Normal	Due date:	
Assignee:	Jean-Philippe Lang	% Done:	0%
Category:	Code cleanup/refactoring	Estimated time:	0.00 hour
Target version:	3.2.5	Affected version:	
Resolution:	Fixed		
Description			
<p>Under certain conditions (and due to Rails' decision to hold global query state on models), under certain conditions, there might be multiple instances of the AnonymousUser and AnonymousGroup being created. Consider this example:</p>			
<pre>User.current # => #<AnonymousUser id: 4 ...> # Create a new public projects with no project members project = Project.create(name: 'Public test', identifier: 'public-test', public: true) # Now, gather the list of visible principals of the project # This is e.g. called by IssueQuery#initialize_available_filters project.principals.visible</pre>			
<p>As you can see, this creates (depending a bit on the settings) a new AnonymousGroup and a new AnonymousUser object. The reason for that with the call above this is called:</p>			
<pre>Principal.active.joins(:members).where("#{Member.table_name}.project_id = ?", project.id).uniq. visible</pre>			
<p>Now, in the visible scope of the Principle, a call to AnonymousUser.first class is used to find the anonymous user (and equivalently the AnonymousGroup later). Unfortunately this happens while the query scope for the outer query is still active. The new and unrelated query for the anonymous user appears to inherits the existing scope on the Principle class of the outer query which results in Redmine not finding the anonymous user and subsequently creating a new one.</p>			
<p>This behavior can be reliably reproduced when browsing as Anonymous a public project without any explicit members.</p>			
<p>The attached patch against current trunk was developed for one of our customers at Planio who experiences this issue. The problem does also occur in all Redmine versions which support the user visibility feature (that is all Redmine 3 version iirc).</p>			
<p>The patch fixes the problem by performing the queries for the global anonymous objects with an explicit call to unscoped which should remove the inherited outer scope. When merging this, it might be a good idea to also provide a migration which removes any erroneous AnonymousUser, AnonymousGroup, NonMemberGroup and AnonymousRole instances in the database.</p>			
<p>(Boy, this was no fun to find and fix...)</p>			

Associated revisions

Revision 16049 - 2016-12-03 09:57 - Jean-Philippe Lang

Use .unscoped when querying and creating builtin objects (#24156).

Patch by Holger Just.

Revision 16093 - 2016-12-20 09:59 - Jean-Philippe Lang

Merged r16049 (#24156).

Revision 16094 - 2016-12-20 09:59 - Jean-Philippe Lang

Merged r16049 (#24156).

History

#1 - 2016-10-26 07:58 - Jan from Planio www.plan.io

- Target version set to Candidate for next minor release

#2 - 2016-11-23 06:08 - Toshi MARUYAMA

- Description updated

#3 - 2016-11-23 06:12 - Toshi MARUYAMA

- Target version changed from Candidate for next minor release to 3.4.0

#4 - 2016-12-03 10:05 - Jean-Philippe Lang

- Category changed from Permissions and roles to Code cleanup/refactoring

- Status changed from New to Resolved

- Assignee set to Jean-Philippe Lang

- Target version changed from 3.4.0 to 3.2.5

- Resolution set to Fixed

Patch committed, thanks.

#5 - 2016-12-20 10:00 - Jean-Philippe Lang

- Status changed from Resolved to Closed

Files

0001-Explicitly-remove-the-potential-global-scope-on-role.patch	2.51 KB	2016-10-24	Holger Just
---	---------	------------	-------------