Redmine - Defect #35174

Internal Error when accessing the home page of some projects

2021-04-27 16:11 - Miguel Moquillon

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:	Projects	**Estimated time:**	0.00 hour
Target version:			
Resolution:	Invalid	**Affected version:**	4.2.1

Description

Since the upgrade from 4.1.2 to 4.2.0 and 4.2.1, when accessing the home page of some of our projects, we have an "Internal Error" page with the following error messages in the log:

```
ActionView::Template::Error (comparison of NilClass with User failed):
    2:   <div class="members box">
    3:     <h3 class="icon icon-group"><%=l(:label_member_plural)%></h3>
    4:     <% @principals_by_role.keys.sort.each do |role| %>
    5:       <p><span class="label"><%= role %>:</span> <%= @principals_by_role[role].sort.collect
{|p| link_to_user p}.join(", ").html_safe %></p>
    6:     <% end %>
    7:   </div>
    8:   <% end %>

app/views/projects/_members_box.html.erb:5:in `sort'
app/views/projects/_members_box.html.erb:5:in `block in _app_views_projects__members_box_html_erb_
__1165424576507768033_82920'
app/views/projects/_members_box.html.erb:4:in `each'
app/views/projects/_members_box.html.erb:4:in `_app_views_projects__members_box_html_erb___1165424
576507768033_82920'
app/views/projects/show.html.erb:132:in `_app_views_projects_show_html_erb___1481690599684784851_8
2880'
lib/redmine/sudo_mode.rb:61:in `sudo_mode'
```

Note: not all the projects are concerned by this problem.

History

#1 - 2021-04-27 20:13 - Marius BĂLTEANU

I cannot reproduce it, can you tell us more about your environment? Please see Submissions.

#2 - 2021-04-27 20:14 - Marius BĂLTEANU

- Status changed from New to Needs feedback

#3 - 2021-04-27 21:36 - Miguel Moquillon

Operating system: Ubuntu 20.04.2 LTS

ruby bin/about gives:

```
sh: 1: svn: not found
sh: 1: hg: not found
sh: 1: cvs: not found
sh: 1: bzr: not found
Environment:
  Redmine version                4.2.1.stable
  Ruby version                   2.7.2-p137 (2020-10-01) [x86_64-linux]
  Rails version                  5.2.5
  Environment                    production
  Database adapter               Mysql2
  Mailer queue                   ActiveJob::QueueAdapters::AsyncAdapter
  Mailer delivery                smtp
```

```
SCM:
  Git                             2.25.1
  Filesystem
Redmine plugins:
  no plugin installed
```

Our redmine instance is served by Apache2 through passenger: Phusion Passenger(R) 6.0.8

For information, redmine is used since 2010 and was updated at each new release by following the page RedmineUpdate

#4 - 2021-04-28 10:04 - Mizuki ISHIKAWA

I implemented this code on the assumption that @principals_by_role[role] is an array containing only User records, but in Redmine where the problem occurred, @principals_by_role[role] seems to have been an array containing user records and nil.

Example:

```
pry(main)> [User.first, nil, User.second].sort
ArgumentError: comparison of NilClass with User failed
from (pry):18:in `sort'
```

I think the problem can be solved by changing the code as follows.

```
diff --git a/app/views/projects/_members_box.html.erb b/app/views/projects/_members_box.html.erb
index e915ab910a..280f2e3aff 100644
--- a/app/views/projects/_members_box.html.erb
+++ b/app/views/projects/_members_box.html.erb
@@ -2,7 +2,7 @@
   <div class="members box">
     <h3 class="icon icon-group"><%=l(:label_member_plural)%></h3>
     <% @principals_by_role.keys.sort.each do |role| %>
-      <p><span class="label"><%= role %>:</span> <%= @principals_by_role[role].sort.collect{|p| link_to_user
p}.join(", ").html_safe %></p>
+      <p><span class="label"><%= role %>:</span> <%= @principals_by_role[role].compact.sort.collect{|p| link_
to_user p}.join(", ").html_safe %></p>
     <% end %>
   </div>
   <% end %>
```

#5 - 2021-04-28 10:18 - Mizuki ISHIKAWA

I don't know why nil is included, I think it needs improvement if it can happen in other environments.

#6 - 2021-04-28 15:53 - Pavel Rosický

Miguel - could you check that you don't have any orphan project members?

```
rails c -e production
Member.where.not(user_id: Principal.select(:id)).any?
```

it should return "false"

this could happen by bypassing Redmine's API and removing a user directly from the database. Any chance you did something like this in the past?

#7 - 2021-04-28 16:17 - Miguel Moquillon

Pavel Rosický wrote:

> Miguel - could you check that you don't have any orphan project members?
> [...]
> it should return "false"
>
> this could happen by bypassing Redmine's API and removing a user directly from the database. Any chance you did something like this in the past?

Unfortunately, it returns ... "true".
No, we don't use the Redmine's API to do admin tasks (or it should have been a long time ago). What does mean "orphan project member"? A user that is yet a member of a project but whose account has been deleted?

#8 - 2021-04-28 16:28 - Miguel Moquillon

Ok, by issuing

```
Member.select(:user_id).where.not(user_id: Principal.select(:id)).distinct
```

I found the identifier of the user causing the error (I'm not a ruby programmer but I figured out the instruction from your code snippet): there is only one user (an old one).

Can I remove it from the membership of the projects without any consequences by executing the following code (is the code correct?):

```
Member.delete.where(user_id: 6)
```

#9 - 2021-04-28 16:33 - Pavel Rosický

if you remove a user from the system, all dependencies like project members are deleted via callbacks
https://github.com/redmine/redmine/blob/master/app/models/principal.rb#L31

however, if you bypass the API, you end up in a situation there's a member entry that refers to a user which doesn't exist in the `users` table.

Mizuki ISHIKAWA's solution is only fixing the consequences of the database inconsistency.

btw: foreign key constraints in the schema should prohibit such actions on the database level, but Redmine doesn't use them, see
https://www.redmine.org/boards/4/topics/65243

#10 - 2021-04-28 16:37 - Miguel Moquillon

Pavel Rosický wrote:

> if you remove a user from the system, all dependencies like project members are deleted via callbacks
> https://github.com/redmine/redmine/blob/master/app/models/principal.rb#L31
>
> however, if you bypass the API, you end up in a situation there's a member entry that refers to a user which doesn't exist in the `users` table.
>
> Mizuki ISHIKAWA's solution is only fixing the consequences of the database inconsistency.
>
> btw: foreign key constraints in the schema should prohibit such actions on the database level, but Redmine doesn't use them, see
> https://www.redmine.org/boards/4/topics/65243

Ok. When I wrote we don't use the Redmine's API, I means we use the Redmine administration page for doing such tasks. (In my mind, Redmine API were meaning REST API.)

#11 - 2021-04-28 16:39 - Pavel Rosický

> Can I remove it?

you should do a backup first before such actions. But each member should refer to an existing user.

#12 - 2021-04-28 16:45 - Pavel Rosický

it's hard to guess, how this could happen in your environment. If it used to work in the previous version and you have a backup, try to compare the data, when the user was removed and how...

#13 - 2021-04-29 09:44 - Miguel Moquillon

Pavel Rosický wrote:

> > Can I remove it?
>
> you should do a backup first before such actions. But each member should refer to an existing user.

Backup of our redmine installation (with database) are performed every night. So in the morning I purged the memberships of the deleted old user by issuing the following command:

```
Member.where(user_id: 6).destroy_all
```

According to the Rails documentation, destroy_all triggers the execution of all the callbacks of the objects.

Once done, we haven't anymore the internal error when accessing the projects' home page.

#14 - 2021-04-29 09:47 - Miguel Moquillon

Pavel Rosický wrote:

> it's hard to guess, how this could happen in your environment. If it used to work in the previous version and you have a backup, try to compare the data, when the user was removed and how...

Unfortunately, we don't keep our backups a long time; we purged them every month.

#15 - 2021-06-08 17:16 - Ingo Linde

Same issue occured here after upgrading to 4.2.1. After removing the orphaned member from the database the issue was resolved.
Thanks for the hints to debug and solve this!

In our case the database inconsistency has persisted at least since 2016. So it is definitely NOT a (recent) migration issue or anything like that.
I can only guess, but I reckon that it has been caused by manual database manipulation back then.

#16 - 2021-06-09 01:20 - Marius BĂLTEANU

- Status changed from Needs feedback to Closed

- Resolution set to Invalid

Ingo Linde wrote:

> Same issue occured here after upgrading to 4.2.1. After removing the orphaned member from the database the issue was resolved.
> Thanks for the hints to debug and solve this!
>
> In our case the database inconsistency has persisted at least since 2016. So it is definitely NOT a (recent) migration issue or anything like that.
> I can only guess, but I reckon that it has been caused by manual database manipulation back then.

Thanks Ingo for the info.

I'm closing this issue because these cases can be fixed manually and I don't think that it's a good idea to add a migration to Redmine to delete the orphan members. Please reopen if I'm wrong.

Files

redmine-Internal_error_page.png	13.5 KB	2021-04-27	Miguel Moquillon
redmine-internal_errror_page.log	1.44 KB	2021-04-27	Miguel Moquillon