

Redmine - Defect #11346

Issue.save force Issue.assigned_to from nil to current user

2012-07-08 21:04 - Thomas Kohler

Status:	Closed	Start date:	
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:	Issues	Estimated time:	0.00 hour
Target version:		Affected version:	1.4.3
Resolution:	Cant reproduce		
Description			
<p>Hello, I'm writing a plug in (redmine_import_reqpro) and every time I programmed a Issue.save the assigned to field is filled with a user (not the the current user?). Also this user is not a member of this project (similar was reported on other Defects #7083 / #7237). The actually workaround is to save the assigned_to field before saving and restore it after saving.</p> <p>thanks</p> <ul style="list-style-type: none">• database manager version: mysql 2.5.0 (gentoo 5.1.62-r1)• Ruby version: 1.8.7• Rails version: 2.3.14			

History

#1 - 2012-07-09 00:59 - Toshi MARUYAMA

- Description updated

#2 - 2012-07-09 18:37 - Jean-Philippe Lang

Could you provide a small code snippet that illustrates the problem?

#3 - 2012-07-11 21:45 - Thomas Kohler

Unfortunately the following code don't produce the error above. I run this code in the first view of my application ("index").

```
a_issue = Issue.new
a_issue.subject = "MyTestIssue"
a_issue.description = "MyDescription"
a_issue.author = User.current
a_issue.project = Project.find_by_id(1)
a_issue.tracker = Tracker.find_by_id(1)
debugger # a_issue.assigned_to nil
puts "now save the issue"
a_issue.save
puts "the issue was saved" # a_issue.assigned_to nil like before
debugger
```

#4 - 2012-07-11 22:06 - Thomas Kohler

Now I have all my code portions with an "any_issue.save" replaced with the call of following function:

```
#TODO: new_issue.save force "assigned_to" to a user (only while first save), how and why?
#wg. redmine Defect #11348
def issue_save_with_assignee_restore(a_issue)
  # wegen bug, dass assigned_to zwangsweise bei issue.save angelegt wird
  tmp_a_to = a_issue.assigned_to
  tmp_a_to_id = a_issue.assigned_to_id

  result = a_issue.save

  if a_issue.assigned_to != tmp_a_to
    debugger
    if tmp_a_to != nil
      puts "assignee canged, old: " + tmp_a_to + ", new: " + a_issue.assigned_to
    else
      puts "assignee canged, old: nil, new: " + a_issue.assigned_to.login
    end
  end
end
```

```
end
  debugger
end
```

```
a_issue.assigned_to = tmp_a_to
return result
end
```

And now sometimes the debugger stops the application because "a_issue.assigned_to != tmp_a_to". But it is always the same user (not the current) see below - password and salt deleted and e-mail changed for this report. I have changed the user status to 3 (user locked) but the problem even exists.

```
<User id: 3, login: "tko", hashed_password: "...", firstname: "Thomas", lastname: "Kohler", mail: "t@de", admin: true, status: 3, last_login_on: "2012-07-02 21:07:04", language: "de", auth_source_id: nil, created_on: "2011-04-07 22:14:29", updated_on: "2012-07-11 21:26:30", type: "User", identity_url: nil, mail_notification: "only_my_events", salt: "...">
```

#5 - 2012-07-11 22:20 - Thomas Kohler

Now I have deleted this "bad user" with id=3 ... and the problem is gone. I have added a new user with exactly the same attributes like the old "bad user" and the problem is not come back.

#6 - 2021-06-07 14:54 - Go MAEDA

- *Status changed from New to Closed*
- *Resolution set to Cant reproduce*

Thomas Kohler wrote:

Now I have deleted this "bad user" with id=3 ... and the problem is gone. I have added a new user with exactly the same attributes like the old "bad user" and the problem is not come back.

Close because it is not reproducible.