Redmine - Defect #39186

Missing synchronization between watchers and watcher_users for unsaved objects

2023-10-12 19:26 - Holger Just

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Go MAEDA	**% Done:**	0%
Category:	Issues	**Estimated time:**	0.00 hour
Target version:	5.1.0		
Resolution:	Fixed	**Affected version:**	

Description

(Note: I'm using issues as examples for watched objects throughout this explanation. The bug applies to all other object types too.)

When adding a new watcher to a new (unsaved) issue using add_watcher, we currently create a new (unsaved) Watcher object in the watchers association in Redmine::Acts::Watchable::InstanceMethods#add_watcher. However, this user is not included in the watcher_users association, nor its basic watcher_users_id list.

This can lead to issues when client code attempts to avoid adding the same user multiple times as a watcher as both lists are not synchronized. Consider the following example as a base case. Here, we try to add the same user multiple times as a watcher, which fails with a uniqueness error :

```
issue = Issue.new(project: Project.first, subject: 'Test', author: User.first, tracker: Project.
first.trackers.first)

issue.add_watcher(User.first)
issue.add_watcher(User.first)

issue.save!
# => ActiveRecord::RecordInvalid (Validation failed: Watchers is invalid)

issue.watchers.last.errors.full_messages.to_sentence
# => User has already been taken
```

This is well and expected, as an object can be watched by a single user only once.

Now however, we can cause unexpected problems if we try to avoid this error by merely checking if the supposed user already watched the issue, e.g.:

```
issue = Issue.new(project: Project.first, subject: 'Test', author: User.first, tracker: Project.
first.trackers.first)

issue.add_watcher(User.first)
issue.add_watcher(User.first) unless issue.watched_by?(User.first)

issue.save!
# => ActiveRecord::RecordInvalid (Validation failed: Watchers is invalid)
```

As you can see, we have not solved the issue with a (supposedly) suitable check. Let's have a deeper look:

```
issue = Issue.new(project: Project.first, subject: 'Test', author: User.first, tracker: Project.
first.trackers.first)
issue.add_watcher(User.first)

issue.watchers
# => #<ActiveRecord::Associations::CollectionProxy [#<Watcher id: nil, watchable_type: "Issue", wa
tchable_id: 17, user_id: 1>]>

issue.watcher_user_ids
# => []
```

As you can see, there is a discrepancy here which results in all checks using watcher_user_ids or watcher_users to be incorrect for (yet unsaved) watchers added in the current request. This includes checks like the following two examples:

```
issue.add_watcher(user) unless issue.watched_by?(user)

users = User.active.to_s
users -= issue.watcher_users
users.each do |u|
  obj.add_watcher(u)
end
```

The latter example is adapted from MailHandler#add_watchers, which is where I detected the bug.

Note that this bug only manifests as long as the issue is not persisted yet. If the issue was already persisted, the watcher_users.reset line in Redmine::Acts::Watchable::InstanceMethods#add_watcher works correctly, so that on next access to watcher_users or watcher_user_ids, the updated state is loaded from the database.

Now, to fix this behavior for new objects, I propose to adapt Redmine::Acts::Watchable::InstanceMethods#add_watcher. In case we are dealing with an unsaved object, we will thus not create a Watcher object directly but append to the watcher_users list instead. Here, Rails will implicitly create (and remove) the Watcher object from the (unsaved) watchers list as required and save them when the underlying object (the issue in this case) is saved.

The underlying problem is rather obscure to reproduce with code plain Redmine, as it manifests itself only when a single unsaved object is passed around and multiple places try to add watchers. At Planio, we observed the issue in multiple stages of mail processing where I suspect, this is more likely to occur with plugins.

Attached to this issue, you will find two patches which attempt to solve this issue:

- 0001-Fix-watcher-handling-on-unsaved-objects.patch - This patch solves the bug described above. For persisted objects, the behavior is unchanged.
- 0002-Add-requirement-for-user_preferences-for-watcher_tes.patch - This (partially unrelated) patch adds the :user_preferences fixture requirement to WatcherTest as some tests (at least test_watcher_user_ids_should_make_ids_uniq) rely on the stored user preferences for auto watching in issue create

Associated revisions

Revision 22349 - 2023-10-16 17:03 - Go MAEDA

Fix watcher handling on unsaved objects (#39186).

Patch by Holger Just.

Revision 22350 - 2023-10-16 17:05 - Go MAEDA

Add requirement for user_preferences for watcher_test (#39186).

Patch by Holger Just.

History

#1 - 2023-10-15 05:57 - Mischa The Evil

That's some proper research wrapped-up in a great issue!

Holger Just wrote:

> As you can see, we have not solved the issue with a (supposedly) suitable check.
>
> [...]
>
> As you can see, there is a discrepancy here which results in all checks using watcher_user_ids or watcher_users to be incorrect for (yet unsaved) watchers added in the current request. This includes checks like the following two examples:
>
> [...]

This seems indeed like bad behavior to me too.

Holger Just wrote:

> Now, to fix this behavior for new objects, I propose to adapt Redmine::Acts::Watchable::InstanceMethods#add_watcher. In case we are dealing with an unsaved object, we will thus not create a Watcher object directly but append to the watcher_users list instead. Here, Rails will implicitly create (and remove) the Watcher object from the (unsaved) watchers list as required and save them when the underlying object (the issue in this case) is saved.

This seems like a proper fix for the given problem.

#2 - 2023-10-16 17:06 - Go MAEDA

- Status changed from New to Closed

- Assignee set to Go MAEDA

- Target version set to 5.1.0

- Resolution set to Fixed

Committed the patches in r22349 and r22350. Thank you for fixing the issue.

Files

0001-Fix-watcher-handling-on-unsaved-objects.patch	3.36 KB	2023-10-12	Holger Just
0002-Add-requirement-for-user_preferences-for-watcher_tes.patch	983 Bytes	2023-10-12	Holger Just