

Redmine - Defect #38636

NoMethodError when creating/updating an issue if all issue priorities are disabled and no default priority is set

2023-05-31 07:40 - Go MAEDA

Status:	New	Start date:	
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:	Issues	Estimated time:	0.00 hour
Target version:		Affected version:	
Resolution:			

Description

NoMethodError is raised when creating or updating an issue if all of the following conditions are met:

- All issue priorities are disabled
- No default priority is set

Error when creating an issue:

```
NoMethodError (undefined method `id' for nil:NilClass):  
  
app/models/issue.rb:2053:in `clear_disabled_fields'  
app/controllers/issues_controller.rb:152:in `create'  
lib/redmine/sudo_mode.rb:61:in `sudo_mode'
```

Error when updating an issue:

```
NoMethodError (undefined method `position' for nil:NilClass):  
  
app/models/issue_priority.rb:63:in `high?'  
app/models/issue.rb:1124:in `notified_users'  
app/models/journal.rb:171:in `notified_users'  
app/models/mailer.rb:134:in `deliver_issue_edit'  
app/models/journal.rb:357:in `send_notification'  
app/controllers/issues_controller.rb:652:in `save_issue_with_child_records'  
app/controllers/issues_controller.rb:207:in `update'  
lib/redmine/sudo_mode.rb:61:in `sudo_mode'
```

Related issues:

Related to Redmine - Patch #40194: Fix error in clear_disabled_fields method ...

New

History

#1 - 2023-05-31 07:54 - Go MAEDA

Since the priority_id column of the issues table has a NOT NULL constraint, it should be prohibited to disable all issue priorities.

#2 - 2024-01-25 09:32 - Go MAEDA

It can be fixed by updating issue.rb like this:

```
diff --git a/app/models/issue_priority.rb b/app/models/issue_priority.rb  
index 6cc75b904..a92da9140 100644  
--- a/app/models/issue_priority.rb  
+++ b/app/models/issue_priority.rb  
@@ -20,6 +20,8 @@  
 class IssuePriority < Enumeration  
   has_many :issues, :foreign_key => 'priority_id'  
  
+ validate :at_least_one_active, if: :active_changed?  
+ after_destroy {|priority| priority.class.compute_position_names}
```

```
after_save do |priority|
  if (priority.saved_change_to_position? && priority.position) ||
@@ -88,4 +90,13 @@ class IssuePriority < Enumeration
    end
  end
end
+
+ private
+
+ def at_least_one_active
+   # Check if this record is inactive and no other active records exist
+   if !active && IssuePriority.where(active: true).where.not(id: self.id).none?
+     errors.add(:base, "At least one IssuePriority must be active.")
+   end
+ end
end
```

#3 - 2024-02-06 06:08 - Go MAEDA

- Related to Patch #40194: Fix error in clear_disabled_fields method when IssuePriority is missing added