Redmine - Patch #27695

Fix ActiveRecord::RecordNotUnique errors when trying to add certain issue relations

2017-12-01 16:48 - Holger Just

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Go MAEDA	**% Done:**	0%
Category:	Issues	**Estimated time:**	0.00 hour
Target version:	3.3.6		

Description

When adding issue relations, under certain conditions, it is not possible for the model validations to check whether a new relation can be added. This results in an ActiveRecord::RecordNotUnique error to be thrown due to the UNIQUE index on the database.

There are two cases where this can occur:

The first case are race conditions with multiple quick attempts to add the relation which end up on multiple worker processes and are handled concurrently. This is a common (but generally rather unlikely) issue with all unique validations on models.

The second case (which in this case is a logic issue) is triggered if there already are directed relations between issues and a user tries to add the reverse relation. Consider this case:

- There exists a precedes relation between issues 1 and 2.
- The user tries to add a follows relation between issues 2 and 1
  - After validation but before save, the new relation is normalized in Relation#handle_issue_order to be saved as "1 precedes 2".
  - A save of this normalized relation triggers the uniqueness constraint on the database and results in an unhandled ActiveRecord::RecordNotUnique error

The root cause of the issue is that the normalization happens after validation. This was moved from before_validation to @before_save in r3191. I guess his was done so that validation errors do not alter the displayed form with the validation errors.

The attached patch (which I extracted from Planio) tries to deal with this situation by handling the exception in the controller and to add a generic error to the model. This can handle both cases described above without an unhandled exception by showing a suitable error message to all clients.

Associated revisions

Revision 17141 - 2018-01-07 01:37 - Go MAEDA

Handle validation errors on reverse issue relations (#27695).

Patch by Holger Just.

Revision 17142 - 2018-01-07 02:30 - Go MAEDA

Merged r17141 from trunk to 3.4-stable (#27695).

Revision 17143 - 2018-01-07 02:31 - Go MAEDA

Merged r17141 from trunk to 3.3-stable (#27695).

History

#1 - 2018-01-06 06:24 - Go MAEDA

- Target version set to 3.3.6

I confirmed the problem by trying the second case. Setting target version to 3.3.6.

```
Started POST "/issues/32/relations" for 127.0.0.1 at 2018-01-06 05:09:31 +0000
Processing by IssueRelationsController#create as JS
  Parameters: {"utf8"=>"✓", "relation"=>{"relation_type"=>"follows", "issue_to_i
d"=>"31", "delay"=>""}, "commit"=>"Add", "issue_id"=>"32"}
...
(snip)
...
```

```
Completed 500 Internal Server Error in 90ms (ActiveRecord: 13.8ms)

ActiveRecord::RecordNotUnique (SQLite3::ConstraintException: UNIQUE constraint failed: issue_relations.issue_f
rom_id, issue_relations.issue_to_id: INSERT INTO "issue_relations" ("issue_from_id", "issue_to_id", "relation_
type", "delay") VALUES (?, ?, ?, ?)):

app/controllers/issue_relations_controller.rb:49:in `create'
lib/redmine/sudo_mode.rb:63:in `sudo_mode'
```

#2 - 2018-01-07 01:41 - Go MAEDA

- Status changed from New to Resolved

- Assignee set to Go MAEDA

Committed to the trunk.
Thank you for investigating and fixing this issue.

#3 - 2018-01-07 02:34 - Go MAEDA

- Status changed from Resolved to Closed

Merged the fix to stable branches.

Files

0001-Handle-validation-errors-on-reverse-issue-relations.patch	1.03 KB	2017-12-01	Holger Just