

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

2022/05/09 18:26 - Admin Redmine

ステータス:	Closed	開始日:	2022/05/09
優先度:	通常	期日:	
担当者:		進捗率:	0%
カテゴリ:	Issues_2	予定工数:	0.00時間
対象バージョン:	3.3.6_136	作業時間:	0.00時間
Redmineorg_URL:	https://www.redmine.org/issues/27695	status_id:	5
category_id:	2	tracker_id:	3
version_id:	136	plus1:	0
issue_org_id:	27695	affected_version:	
author_id:	2784	closed_on:	
assigned_to_id:	332	affected_version_id:	
comments:	3		

説明

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 this was done so that validation errors do not alter the displayed form with the validation errors.

The attached patch (which I extracted from "Planio":<https://plan.io/redmine-hosting/>) 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.

journals

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_id"=>"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_from_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'

Committed to the trunk.

Thank you for investigating and fixing this issue.

Merged the fix to stable branches.

履歴

#1 - 2022/05/10 17:05 - Admin Redmine

- カテゴリ を Issues_2 にセット

- 対象バージョン を 3.3.6_136 にセット