

Redmine - Defect #5907

ActionView::TemplateError undefined method `-' for nil:NilClass - missing database migration

2010-07-19 15:24 - Sascha Drews

Status:	Closed	Start date:	2010-07-19
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:	Issues	Estimated time:	0.00 hour
Target version:		Affected version:	
Resolution:	Invalid		

Description

Hi there,

I have problems with issues list in 1.0.0.

I have a working installation of 0.9.5 (about 15 projects, some of them with svn integration, running on port 3000) and wanted to run a test stage for 1.0.0.

The machine is a Ubuntu 10.04 64bit server.

Ruby version 1.8.7 (x86_64-linux)

RubyGems version 1.3.6

Rack version 1.0

Rails version 2.3.5

Active Record version 2.3.5

Active Resource version 2.3.5

Action Mailer version 2.3.5

Active Support version 2.3.5

MySQL Server version 5.1.41

Here's what I did.

I downloaded 1.0.0 and unpacked it to '/srv/rails/redmine-1.0.0' (our working Redmine is under '/srv/rails/redmine-0.9.5'), duplicated the db (from 'redmine' to 'redmine-stage'), copied database.yml and email.yml, edited database.yml to point to the duplicated db and copied the 'files' folder.

After that I ran (in 1.0.0):

```
rake generate_session_store
```

```
rake tmp:cache:clear
```

```
rake tmp:sessions:clear
```

Finally I started the server with:

```
ruby script/server webrick -e production -p 3001
```

Calling the 'issues' tab for any project gives me an 500-error, production.log says:

```
Processing IssuesController#index (for 172.X.X.X at 2010-07-19 14:12:24) [GET]
  Parameters: {"project_id"=>"project42", "action"=>"index", "controller"=>"issues"}
```

```
Rendering template within layouts/base
```

```
Rendering issues/index.rhtml
```

```
ActionView::TemplateError (undefined method `-' for nil:NilClass) on line #16 of app/views/issues/_list.rhtml:
```

```
13:     </tr></thead>
```

```
14:     <% previous_group = false %>
```

```
15:     <tbody>
```

```
16:     <% issue_list(issues) do |issue, level| -%>
```

```
17:     <% if @query.grouped? && (group = @query.group_by_column.value(issue)) != previous_group %>
```

```
18:     <% reset_cycle %>
```

```
19: <tr class="group open">
  app/helpers/issues_helper.rb:28:in `issue_list'
  app/helpers/issues_helper.rb:23:in `each'
  app/helpers/issues_helper.rb:23:in `issue_list'
  app/views/issues/_list.rhtml:16
  app/views/issues/_list.rhtml:1:in `_run_rhtml_app47views47issues47_list46rhtml_locals_issues_l
ist_object_query'
  app/views/issues/index.rhtml:62:in `_run_rhtml_app47views47issues47index46rhtml'
  app/controllers/issues_controller.rb:83
  app/controllers/issues_controller.rb:82:in `index'
  /usr/lib/ruby/1.8/webrick/httpserver.rb:104:in `service'
  /usr/lib/ruby/1.8/webrick/httpserver.rb:65:in `run'
  /usr/lib/ruby/1.8/webrick/server.rb:173:in `start_thread'
  /usr/lib/ruby/1.8/webrick/server.rb:162:in `start'
  /usr/lib/ruby/1.8/webrick/server.rb:162:in `start_thread'
  /usr/lib/ruby/1.8/webrick/server.rb:95:in `start'
  /usr/lib/ruby/1.8/webrick/server.rb:92:in `each'
  /usr/lib/ruby/1.8/webrick/server.rb:92:in `start'
  /usr/lib/ruby/1.8/webrick/server.rb:23:in `start'
  /usr/lib/ruby/1.8/webrick/server.rb:82:in `start'
```

Rendering /srv/rails/redmine-1.0.0/public/500.html (500 Internal Server Error)

Can anyone help me?

Sascha

Related issues:

Has duplicate Redmine - Defect #7088: 500 Internal Error: when click "issues"...	Closed	2010-12-10
--	--------	------------

History

#1 - 2010-07-19 16:25 - Guilherme Schneider

I guess you're missing rake db:migrate, no?

#2 - 2010-07-19 16:32 - Sascha Drews

Oh, I'm sorry, I forgot to mention that - of course I also ran rake db:migrate.
I did everything that is stated in [RedmineUpgrade](#), except for the plugins instructions, since I have none installed.

Running ruby script/about tells me:
Database schema version 20100705164950

Hope this helps.
Sascha

#3 - 2010-07-19 23:21 - Felix Schäfer

If you had the subtasks plugin installed: it's not compatible with 1.0.0, and especially not the data format. I think there was a way to revert the migrations from a plugin (something like `RAILS_ENV=production rake db:migrate:plugins plugin_name version=0` or similar), if you want an upgrade-path without loss of data, contact the author of the plugin.

#4 - 2010-07-20 04:32 - Stepan Volkov

Felix Schäfer wrote:

If you had the subtasks plugin installed: it's not compatible with 1.0.0, and especially not the data format. I think there was a way to revert the migrations from a plugin (something like `RAILS_ENV=production rake db:migrate:plugins plugin_name version=0` or similar), if you want an upgrade-path without loss of data, contact the author of the plugin.

I have some problem and resolve it in <http://www.redmine.org/boards/2/topics/15729#message-15734>

I not install subtask plugin, my way to migrate error in my hand:

- 1) some time ago I up my redmine from trunk
- 2) then I switch to 0.9-stable

on migrate from 0.9 to 1.0 I have promlem with database changes (column already exist)

#5 - 2010-07-20 09:51 - Sascha Drews

Felix Schäfer wrote:

If you had the subtasks plugin installed: it's not compatible with 1.0.0, and especially not the data format. I think there was a way to revert the migrations from a plugin (something like `RAILS_ENV=production rake db:migrate:plugins plugin_name version=0` or similar), if you want an upgrade-path without loss of data, contact the author of the plugin.

But I had no plugins installed, neither the subtasks plugin nor any other plugin... :-(

#6 - 2010-07-20 10:22 - Sascha Drews

Stepan Volkov wrote:

Felix Schäfer wrote:

If you had the subtasks plugin installed: it's not compatible with 1.0.0, and especially not the data format. I think there was a way to revert the migrations from a plugin (something like `RAILS_ENV=production rake db:migrate:plugins plugin_name version=0` or similar), if you want an upgrade-path without loss of data, contact the author of the plugin.

I have some problem and resolve it in <http://www.redmine.org/boards/2/topics/15729#message-15734>

I not install subtask plugin, my way to migrate error in my hand:

- 1) some time ago I up my redmine from trunk
- 2) then I switch to 0.9-stable

on migrate from 0.9 to 1.0 I have problem with database changes (column already exist)

I gave the suggestions from Stepan a try, although I'm not on PostgreSQL but MySQL and I had no errors while migrating.

- First I dropped my stage-db and recreated it from dump
- Then I dropped columns ``parent_id``, ``lft`` and ``rgt`` of table ``issues``
- `ruby script/about` tells me Database schema version 20100313171051
- Next `rake db:migrate RAILS_ENV=production`

```
(in /srv/rails/redmine-1.0.0)
== ChangeChangesPathLengthLimit: migrating =====
-- change_column(:changes, :path, :text, {:null=>false})
  -> 0.3050s
-- change_column(:changes, :from_path, :text)
  -> 0.4371s
== ChangeChangesPathLengthLimit: migrated (0.7426s) =====
```

- `ruby script/about` tells me Database schema version 20100705164950
- For the sake of completeness I ran `rake db:migrate:upgrade_plugin_migrations RAILS_ENV=production`

```
(in /srv/rails/redmine-1.0.0)
Cannot find old migration table - assuming nothing needs to be done
```

- and `rake db:migrate_plugins RAILS_ENV=production`

```
(in /srv/rails/redmine-1.0.0)
Migrating engines...
Migrating acts_as_activity_provider...
Migrating acts_as_attachable...
Migrating acts_as_customizable...
Migrating acts_as_event...
Migrating acts_as_list...
Migrating acts_as_searchable...
Migrating acts_as_tree...
Migrating acts_as_versioned...
Migrating acts_as_watchable...
Migrating awesome_nested_set...
Migrating classic_pagination...
Migrating coderay-0.9.2...
Migrating gravatar...
Migrating open_id_authentication...
Migrating prepend_engine_views...
Migrating rfpdf...
Migrating ruby-net-ldap-0.0.4...
```

- Finally some rake tmp:cache:clear and rake tmp:sessions:clear
- And then, big surprise, after I started the server again I see...

```
Processing IssuesController#index (for 172.X.X.X at 2010-07-20 10:05:47) [GET]
  Parameters: {"project_id"=>"project42", "set_filter"=>"1", "action"=>"index", "controller"=>"issues"}
Rendering template within layouts/base
Rendering issues/index.rhtml

ActionView::TemplateError (undefined method `-' for nil:NilClass) on line #16 of app/views/issues/_list.rhtml:
13:     </tr></thead>
14:     <% previous_group = false %>
15:     <tbody>
16:     <% issue_list(issues) do |issue, level| -%>
17:     <% if @query.grouped? && (group = @query.group_by_column.value(issue)) != previous_group %>
18:     <% reset_cycle %>
19:     <tr class="group open">

app/helpers/issues_helper.rb:28:in `issue_list'
app/helpers/issues_helper.rb:23:in `each'
app/helpers/issues_helper.rb:23:in `issue_list'
app/views/issues/_list.rhtml:16
app/views/issues/_list.rhtml:1:in `_run_rhtml_app47views47issues47_list46rhtml_locals_issues_list_object_query'
app/views/issues/index.rhtml:62:in `_run_rhtml_app47views47issues47index46rhtml'
app/controllers/issues_controller.rb:83
app/controllers/issues_controller.rb:82:in `index'
/usr/lib/ruby/1.8/webrick/httpserver.rb:104:in `service'
/usr/lib/ruby/1.8/webrick/httpserver.rb:65:in `run'
/usr/lib/ruby/1.8/webrick/server.rb:173:in `start_thread'
/usr/lib/ruby/1.8/webrick/server.rb:162:in `start'
/usr/lib/ruby/1.8/webrick/server.rb:162:in `start_thread'
/usr/lib/ruby/1.8/webrick/server.rb:95:in `start'
/usr/lib/ruby/1.8/webrick/server.rb:92:in `each'
/usr/lib/ruby/1.8/webrick/server.rb:92:in `start'
/usr/lib/ruby/1.8/webrick/server.rb:23:in `start'
/usr/lib/ruby/1.8/webrick/server.rb:82:in `start'

Rendering /srv/rails/redmine-1.0.0/public/500.html (500 Internal Server Error)
```

:-)

#7 - 2010-07-20 12:07 - Stepan Volkov

Sascha Drews wrote:

- First I dropped my stage-db an recreated it from dump
- Then I dropped columns `parent_id`, `lft` and `rgt` of table `issues`
- ruby script/about tells me Database schema version 20100313171051
- Next rake db:migrate RAILS_ENV=production

I think your way small wrong. Need drop columns if (and only if) db:migrate tell you about any of column already exists in table issues.

Try:

- 1) drop my stage-db an recreate it from dump
- 2) run rake db:migrate RAILS_ENV=production
- 3) see error messages like:

```
-- add_column(:issues, :parent_id, :integer, {:default=>nil})
rake aborted!
An error has occurred, this and all later migrations canceled:

PGError: ERROR: column "parent_id" of relation "issues" already exists
: ALTER TABLE "issues" ADD COLUMN "parent_id" integer
```

- 4) if you see error message already exists drop column, and try db:migrate (goto 2)

your db:migrate to short:

```
== ChangeChangesPathLengthLimit: migrating =====
```

```
-- change_column(:changes, :path, :text, {:null=>false})
-> 0.3050s
-- change_column(:changes, :from_path, :text)
-> 0.4371s
== ChangeChangesPathLengthLimit: migrated (0.7426s) =====
```

if you drop any column it can be added after run rake db:migrate.
My log db:migrate after drop (conflicted columns in my DB) `parent_id`, `lft` and `rgt` on table `issues`:

```
== AddIssuesNestedSetsColumns: migrating =====
-- add_column(:issues, :parent_id, :integer, {:default=>nil})
-> 0.0203s
-- add_column(:issues, :root_id, :integer, {:default=>nil})
-> 0.0019s
-- add_column(:issues, :lft, :integer, {:default=>nil})
-> 0.0018s
-- add_column(:issues, :rgt, :integer, {:default=>nil})
-> 0.0018s
== AddIssuesNestedSetsColumns: migrated (0.1846s) =====

== AddIndexOnIssuesNestedSet: migrating =====
-- add_index(:issues, [:root_id, :lft, :rgt])
-> 0.0058s
== AddIndexOnIssuesNestedSet: migrated (0.0060s) =====

== ChangeChangesPathLengthLimit: migrating =====
-- change_column(:changes, :path, :text, {:null=>false})
-> 0.2177s
-- change_column(:changes, :from_path, :text)
-> 0.2101s
== ChangeChangesPathLengthLimit: migrated (0.4283s) =====
```

`parent_id`, `lft` and `rgt` restored.

Drop only conflicted (exists) columns!

#8 - 2010-07-21 10:28 - Sascha Drews

Good news everyone, I could solve the problem.

After reading [Felix' comment](#) I took another close look at my db.

I saw that I have a lot of records in my issues table that don't have the columns **root_id**, **lft**, **rgt** filled.

id	...	parent_id	root_id	lft	rgt
562	...	NULL	562	1	2
563	...	NULL	563	1	2
564	...	NULL	564	1	2
565	...	NULL	NULL	NULL	NULL
566	...	NULL	NULL	NULL	NULL
567	...	NULL	NULL	NULL	NULL

So I gave this simple sql statement a chance:

```
update issues set parent_id = NULL, root_id = id, lft = 1, rgt = 2;
```

I can run that on all records without any conditions since I had no subtasks before so I had not to worry about damaging existing relations.

And well, after that I can see my issues lists in all of my projects.

I think, some kind of update like I did should be included in the migration scripts...

Thanks Felix for pointing me in the right direction :-)

#9 - 2010-07-21 11:01 - Felix Schäfer

- Status changed from New to Closed

Ok, closing this as it seems to have been a glitch in you upgrade process.

#10 - 2010-08-31 01:39 - love_obe love_obe

- Status changed from Closed to Reopened

I got a same problem.

Migrating from 0.9.4 to 1.0.1(both are BitNami Redmine Stack).

Badly, I have no parent_id, root_id, lft, rgh in my issues table.
So Sascha's workaround doesn't help me.

By the way, [#6032](#) and [#6193](#) reported the same problem.
I think Redmine 1.0 has the migrating bug.

Does anyone have suggestions?

#11 - 2010-08-31 11:56 - Sergey Belov

Thanks! this DB updated helped!

#12 - 2010-08-31 13:30 - Felix Schäfer

love_obe love_obe wrote:

I got a same problem.

Migrating from 0.9.4 to 1.0.1(both are BitNami Redmine Stack).

Badly, I have no parent_id, root_id, lft, rgh in my issues table.
So Sascha's workaround doesn't help me.

By the way, [#6032](#) and [#6193](#) reported the same problem.
I think Redmine 1.0 has the migrating bug.

Does anyone have suggestions?

Make sure you have migrated correctly: If you have no parent_id, root_id, lft and rgt columns in your schema, the migration wasn't successful.

#13 - 2010-09-01 01:53 - love_obe love_obe

Thanks, Felix.

I redo <http://www.redmine.org/wiki/redmine/RedmineUpgrade#Step-4-Update-the-database> and it works now.

#14 - 2010-09-01 02:03 - Eric Davis

- Subject changed from Cannot use issues list in 1.0.0 to ActionView::TemplateError undefined method `-' for nil:NilClass - missing database migration

- Status changed from Reopened to Closed

- Resolution set to Invalid

#15 - 2016-02-14 19:10 - Raigedas Unknown

In my case the cause was that I have used settings file from older version.