

Redmine - Defect #11905

Potential "can't dup NilClass" error in UserPreference

2012-09-21 14:45 - Karim Helal

Status:	Closed	Start date:	
Priority:	High	Due date:	
Assignee:	Jean-Philippe Lang	% Done:	0%
Category:	Code cleanup/refactoring	Estimated time:	0.00 hour
Target version:	2.1.1	Affected version:	2.1.0
Resolution:	Fixed		
Description			
<p>Hello, we've recently upgraded from 1.4.4 to 2.1.0 and now we get the following error 500 when trying to open up any page:</p> <pre>Connecting to database specified by database.yml OpenIdAuthentication.store is nil. Using in-memory store. Started GET "/login?back_url=http%3A%2F%2Fred.nuserv.com%2F" for 217.164.33.239 at 2012-09-21 12:28:37 +0000 Processing by AccountController#login as HTML Parameters: {"back_url"=>"http://red.nuserv.com/"} Current user: anonymous Rendered account/login.html.erb within layouts/base (11079.3ms) Completed 500 Internal Server Error in 11387ms ActionView::Template::Error (can't dup NilClass): 9: <%= favicon %> 10: <%= stylesheet_link_tag 'jquery/jquery-ui-1.8.21', 'application', :media => 'all' %> 11: <%= stylesheet_link_tag 'rtl', :media => 'all' if l(:direction) == 'rtl' %> 12: <%= javascript_heads %> 13: <%= heads_for_theme %> 14: <%= call_hook :view_layouts_base_html_head %> 15: <!-- page specific tags --> app/models/user_preference.rb:47:in `dup' app/models/user_preference.rb:47:in `[]=' app/models/user_preference.rb:27:in `initialize' app/models/user.rb:664:in `new' app/models/user.rb:664:in `pref' app/helpers/application_helper.rb:1222:in `javascript_heads' app/views/layouts/base.html.erb:12:in `_app_views_layouts_base_html_erb___3022056590706844150_38096160'</pre>			
<p>Any ideas what could be causing this? We are using Ruby 1.9.3 on Ubuntu 12.04.1</p> <p>Thx.</p>			

Associated revisions

Revision 10438 - 2012-09-22 08:16 - Jean-Philippe Lang

Potential can't dup NilClass error in UserPreference (#11905).

History

#1 - 2012-09-22 08:20 - Jean-Philippe Lang

- Subject changed from Problem when upgrading from Redmine 1.4.4 to 2.1.0 to can't dup NilClass error after upgrading to 2.1.0
- Status changed from New to Resolved
- Assignee set to Jean-Philippe Lang
- Target version set to 2.1.1
- Resolution set to Fixed

I never experienced this error when using 2.1.0. But I was able to write a test that triggers this error. It's fixed in r10438. The patch is pretty simple, you can apply it to your 2.1.0 instance then restart:

Index: user_preference.rb

```
=====
--- user_preference.rb (revision 10437)
+++ user_preference.rb (revision 10438)
@@ -44,7 +44,7 @@
   if attribute_present? attr_name
     super
   else
-   h = read_attribute(:others).dup || {}
+   h = (read_attribute(:others) || {}).dup
     h.update(attr_name => value)
     write_attribute(:others, h)
     value
```

#2 - 2012-09-22 09:40 - Karim Helal

Hi Jean-Philippe,

Applied the patch and now get a different set of errors.

First time I tried to login:

Started GET "/login?back_url=http%3A%2F%2Fred.nuserv.com%2F" for 217.164.33.239 at 2012-09-22 07:36:03 +0000

Processing by AccountController#login as HTML

Parameters: {"back_url"=>"http://red.nuserv.com/"}

Current user: anonymous

Rendered account/login.html.erb within layouts/base (11178.9ms)

Rendered plugins/redmine_wiki_extensions/app/views/wiki_extensions/_html_header.html.erb (0.8ms)

Rendered plugins/redmine_wiki_extensions/app/views/wiki_extensions/_body_bottom.html.erb (0.7ms)

Completed 200 OK in 11441ms (Views: 11432.2ms | ActiveRecord: 5.3ms)

Started POST "/login" for 217.164.33.239 at 2012-09-22 07:36:24 +0000

Processing by AccountController#login as HTML

Parameters: {"utf8"=>"", "authenticity_token"=>"TLujlM0YbPGi16LGXwEkm4Q2h2hRSd/CnrN98lk/zt0=",

"back_url"=>"http://red.nuserv.com/", "username"=>"khalal", "password"=>"[FILTERED]", "autologin"=>"1", "login"=>"Login »"}

Current user: anonymous

Successful authentication for 'khalal' from 217.164.33.239 at 2012-09-22 07:36:25 UTC

Completed 500 Internal Server Error in 347ms

ActiveRecord::StatementInvalid (PG::Error: ERROR: null value in column "id" violates not-null constraint

```
: INSERT INTO "tokens" ("action", "created_on", "user_id", "value") VALUES ($1, $2, $3, $4) RETURNING "id":  
app/controllers/account_controller.rb:225:in `set_autologin_cookie'  
app/controllers/account_controller.rb:218:in `successful_authentication'  
app/controllers/account_controller.rb:167:in `password_authentication'  
app/controllers/account_controller.rb:154:in `authenticate_user'  
app/controllers/account_controller.rb:30:in `login'
```

And when trying again:

```
Started GET "/login" for 217.164.33.239 at 2012-09-22 07:37:03 +0000  
Processing by AccountController#login as HTML  
Current user: anonymous  
Rendered account/login.html.erb within layouts/base (1.5ms)  
Rendered plugins/redmine_wiki_extensions/app/views/wiki_extensions/_html_header.html.erb (0.0ms)  
Rendered plugins/redmine_wiki_extensions/app/views/wiki_extensions/_body_bottom.html.erb (0.0ms)  
Completed 200 OK in 9ms (Views: 5.5ms | ActiveRecord: 1.0ms)
```

```
Started POST "/login" for 217.164.33.239 at 2012-09-22 07:37:11 +0000  
Processing by AccountController#login as HTML  
Parameters: {"utf8"=>"", "authenticity_token"=>"TLujlM0YbPGi16LGXwEkm4Q2h2hRSd/CnrN98lk/zt0=", "username"=>"khelal",  
"password"=>"[FILTERED]", "autologin"=>"1", "login"=>"Login »"}  
Current user: anonymous  
Failed login for 'khelal' from 217.164.33.239 at 2012-09-22 07:37:11 UTC  
Rendered account/login.html.erb within layouts/base (1.2ms)  
Rendered plugins/redmine_wiki_extensions/app/views/wiki_extensions/_html_header.html.erb (0.0ms)  
Rendered plugins/redmine_wiki_extensions/app/views/wiki_extensions/_body_bottom.html.erb (0.0ms)  
Completed 200 OK in 9ms (Views: 4.7ms | ActiveRecord: 1.3ms)
```

```
Started POST "/login" for 217.164.33.239 at 2012-09-22 07:37:17 +0000  
Processing by AccountController#login as HTML  
Parameters: {"utf8"=>"", "authenticity_token"=>"TLujlM0YbPGi16LGXwEkm4Q2h2hRSd/CnrN98lk/zt0=",  
"back_url"=>"http://red.nuserv.com/login", "username"=>"khelal", "password"=>"[FILTERED]", "login"=>"Login »"}  
Current user: anonymous  
Successful authentication for 'khelal' from 217.164.33.239 at 2012-09-22 07:37:17 UTC  
Redirected to http://red.nuserv.com/my/page  
Completed 302 Found in 10ms (ActiveRecord: 4.2ms)
```

```
Started GET "/my/page" for 217.164.33.239 at 2012-09-22 07:37:17 +0000  
Processing by MyController#page as HTML  
Current user: khelal (id=1)  
Completed 500 Internal Server Error in 25ms
```

```
TypeError (can't convert Symbol into Integer):  
app/models/user_preference.rb:39:in `[]'  
app/models/user_preference.rb:39:in `[]'  
app/controllers/my_controller.rb:46:in `page'
```

#3 - 2012-09-22 10:14 - Jean-Philippe Lang

I'm sorry but there's something seriously wrong with your database:

ActiveRecord::StatementInvalid (PG::Error: ERROR: null value in column "id" violates not-null constraint
: INSERT INTO "tokens" ("action", "created_on", "user_id", "value") VALUES (\$1, \$2, \$3, \$4) RETURNING "id"):

There's no way to fix this error in Redmine, it's due to an invalid table definition (the id column should be autoincremented). Make sure to properly restore a database backup that used to work.

#4 - 2012-09-22 10:41 - Karim Helal

Just to be on the safe side, i've re-transferred all the redmine data from our old server (mysql) to the new one (postgresql) using NaviCat Data Transfer and all went well:

```
[Msg] [Dtf] Datatransfer started
[Msg] [Dtf] Gather Information
[Msg] [Dtf] Lock source server tables for read
[Msg] [Dtf] Drop table: "attachments"
[Msg] [Dtf] Get table data for attachments
[Msg] [Dtf] Create table: "attachments"
[Msg] [Dtf] Begin transaction on target server
[Msg] [Dtf] Start transfer data for table: attachments
[Msg] [Dtf] End transaction on target server
[Msg] [Dtf] Drop table: "auth_sources"
[Msg] [Dtf] Get table data for auth_sources
[Msg] [Dtf] Create table: "auth_sources"
[Msg] [Dtf] Drop table: "boards"
[Msg] [Dtf] Get table data for boards
[Msg] [Dtf] Create table: "boards"
[Msg] [Dtf] Drop table: "burndown_days"
[Msg] [Dtf] Get table data for burndown_days
[Msg] [Dtf] Create table: "burndown_days"
[Msg] [Dtf] Drop table: "changes"
[Msg] [Dtf] Get table data for changes
[Msg] [Dtf] Create table: "changes"
[Msg] [Dtf] Begin transaction on target server
[Msg] [Dtf] Start transfer data for table: changes
[Msg] [Dtf] Get table data for changeset_parents
[Msg] [Dtf] End transaction on target server
[Msg] [Dtf] Drop table: "changeset_parents"
[Msg] [Dtf] Create table: "changeset_parents"
[Msg] [Dtf] Begin transaction on target server
[Msg] [Dtf] Start transfer data for table: changeset_parents
[Msg] [Dtf] Get table data for changesets
[Msg] [Dtf] End transaction on target server
[Msg] [Dtf] Drop table: "changesets"
[Msg] [Dtf] Create table: "changesets"
[Msg] [Dtf] Begin transaction on target server
[Msg] [Dtf] Start transfer data for table: changesets
[Msg] [Dtf] Get table data for changesets_issues
[Msg] [Dtf] Get table data for code_categories
[Msg] [Dtf] End transaction on target server
[Msg] [Dtf] Drop table: "changesets_issues"
```

[Msg] [Dtf] Create table: "changesets_issues"
[Msg] [Dtf] Begin transaction on target server
[Msg] [Dtf] Start transfer data for table: changesets_issues
[Msg] [Dtf] End transaction on target server
[Msg] [Dtf] Drop table: "code_categories"
[Msg] [Dtf] Get table data for code_documents
[Msg] [Dtf] Create table: "code_categories"
[Msg] [Dtf] Begin transaction on target server
[Msg] [Dtf] Start transfer data for table: code_categories
[Msg] [Dtf] End transaction on target server
[Msg] [Dtf] Drop table: "code_documents"
[Msg] [Dtf] Create table: "code_documents"
[Msg] [Dtf] Begin transaction on target server
[Msg] [Dtf] Start transfer data for table: code_documents
[Msg] [Dtf] End transaction on target server
[Msg] [Dtf] Drop table: "comments"
[Msg] [Dtf] Get table data for comments
[Msg] [Dtf] Create table: "comments"
[Msg] [Dtf] Drop table: "contacts"
[Msg] [Dtf] Get table data for contacts
[Msg] [Dtf] Create table: "contacts"
[Msg] [Dtf] Begin transaction on target server
[Msg] [Dtf] Start transfer data for table: contacts
[Msg] [Dtf] End transaction on target server
[Msg] [Dtf] Drop table: "contacts_deals"
[Msg] [Dtf] Get table data for contacts_deals
[Msg] [Dtf] Create table: "contacts_deals"
[Msg] [Dtf] Drop table: "contacts_issues"
[Msg] [Dtf] Get table data for contacts_issues
[Msg] [Dtf] Create table: "contacts_issues"
[Msg] [Dtf] Drop table: "contacts_projects"
[Msg] [Dtf] Get table data for contacts_projects
[Msg] [Dtf] Create table: "contacts_projects"
[Msg] [Dtf] Begin transaction on target server
[Msg] [Dtf] Start transfer data for table: contacts_projects
[Msg] [Dtf] End transaction on target server
[Msg] [Dtf] Drop table: "contacts_settings"
[Msg] [Dtf] Get table data for contacts_settings
[Msg] [Dtf] Create table: "contacts_settings"
[Msg] [Dtf] Drop table: "custom_fields"
[Msg] [Dtf] Get table data for custom_fields
[Msg] [Dtf] Create table: "custom_fields"
[Msg] [Dtf] Begin transaction on target server
[Msg] [Dtf] Start transfer data for table: custom_fields
[Msg] [Dtf] End transaction on target server
[Msg] [Dtf] Drop table: "custom_fields_projects"
[Msg] [Dtf] Get table data for custom_fields_projects
[Msg] [Dtf] Create table: "custom_fields_projects"
[Msg] [Dtf] Begin transaction on target server
[Msg] [Dtf] Start transfer data for table: custom_fields_projects
[Msg] [Dtf] End transaction on target server
[Msg] [Dtf] Drop table: "custom_fields_trackers"
[Msg] [Dtf] Get table data for custom_fields_trackers

[Msg] [Dtf] Create table: "custom_fields_trackers"
[Msg] [Dtf] Begin transaction on target server
[Msg] [Dtf] Start transfer data for table: custom_fields_trackers
[Msg] [Dtf] End transaction on target server
[Msg] [Dtf] Drop table: "custom_values"
[Msg] [Dtf] Get table data for custom_values
[Msg] [Dtf] Create table: "custom_values"
[Msg] [Dtf] Begin transaction on target server
[Msg] [Dtf] Start transfer data for table: custom_values
[Msg] [Dtf] Get table data for deal_categories
[Msg] [Dtf] Get table data for deal_processes
[Msg] [Dtf] End transaction on target server
[Msg] [Dtf] Drop table: "deal_categories"
[Msg] [Dtf] Create table: "deal_categories"
[Msg] [Dtf] Drop table: "deal_processes"
[Msg] [Dtf] Get table data for deal_statuses
[Msg] [Dtf] Create table: "deal_processes"
[Msg] [Dtf] Drop table: "deal_statuses"
[Msg] [Dtf] Create table: "deal_statuses"
[Msg] [Dtf] Begin transaction on target server
[Msg] [Dtf] Start transfer data for table: deal_statuses
[Msg] [Dtf] End transaction on target server
[Msg] [Dtf] Drop table: "deal_statuses_projects"
[Msg] [Dtf] Get table data for deal_statuses_projects
[Msg] [Dtf] Create table: "deal_statuses_projects"
[Msg] [Dtf] Drop table: "deals"
[Msg] [Dtf] Get table data for deals
[Msg] [Dtf] Create table: "deals"
[Msg] [Dtf] Drop table: "diagrams"
[Msg] [Dtf] Get table data for diagrams
[Msg] [Dtf] Create table: "diagrams"
[Msg] [Dtf] Drop table: "documentations"
[Msg] [Dtf] Get table data for documentations
[Msg] [Dtf] Create table: "documentations"
[Msg] [Dtf] Drop table: "documents"
[Msg] [Dtf] Get table data for documents
[Msg] [Dtf] Create table: "documents"
[Msg] [Dtf] Drop table: "enabled_modules"
[Msg] [Dtf] Get table data for enabled_modules
[Msg] [Dtf] Create table: "enabled_modules"
[Msg] [Dtf] Begin transaction on target server
[Msg] [Dtf] Start transfer data for table: enabled_modules
[Msg] [Dtf] End transaction on target server
[Msg] [Dtf] Drop table: "enumerations"
[Msg] [Dtf] Get table data for enumerations
[Msg] [Dtf] Create table: "enumerations"
[Msg] [Dtf] Begin transaction on target server
[Msg] [Dtf] Start transfer data for table: enumerations
[Msg] [Dtf] End transaction on target server
[Msg] [Dtf] Drop table: "groups_users"
[Msg] [Dtf] Get table data for groups_users
[Msg] [Dtf] Create table: "groups_users"
[Msg] [Dtf] Begin transaction on target server

[Msg] [Dtf] Start transfer data for table: groups_users
[Msg] [Dtf] End transaction on target server
[Msg] [Dtf] Drop table: "issue_categories"
[Msg] [Dtf] Get table data for issue_categories
[Msg] [Dtf] Create table: "issue_categories"
[Msg] [Dtf] Drop table: "issue_relations"
[Msg] [Dtf] Get table data for issue_relations
[Msg] [Dtf] Create table: "issue_relations"
[Msg] [Dtf] Begin transaction on target server
[Msg] [Dtf] Start transfer data for table: issue_relations
[Msg] [Dtf] End transaction on target server
[Msg] [Dtf] Drop table: "issue_statuses"
[Msg] [Dtf] Get table data for issue_statuses
[Msg] [Dtf] Create table: "issue_statuses"
[Msg] [Dtf] Begin transaction on target server
[Msg] [Dtf] Start transfer data for table: issue_statuses
[Msg] [Dtf] End transaction on target server
[Msg] [Dtf] Drop table: "issues"
[Msg] [Dtf] Get table data for issues
[Msg] [Dtf] Create table: "issues"
[Msg] [Dtf] Begin transaction on target server
[Msg] [Dtf] Start transfer data for table: issues
[Msg] [Dtf] Get table data for journal_details
[Msg] [Dtf] End transaction on target server
[Msg] [Dtf] Get table data for journals
[Msg] [Dtf] Drop table: "journal_details"
[Msg] [Dtf] Create table: "journal_details"
[Msg] [Dtf] Begin transaction on target server
[Msg] [Dtf] Start transfer data for table: journal_details
[Msg] [Dtf] Get table data for member_roles
[Msg] [Dtf] End transaction on target server
[Msg] [Dtf] Drop table: "journals"
[Msg] [Dtf] Create table: "journals"
[Msg] [Dtf] Begin transaction on target server
[Msg] [Dtf] Start transfer data for table: journals
[Msg] [Dtf] Get table data for members
[Msg] [Dtf] End transaction on target server
[Msg] [Dtf] Drop table: "member_roles"
[Msg] [Dtf] Create table: "member_roles"
[Msg] [Dtf] Begin transaction on target server
[Msg] [Dtf] Start transfer data for table: member_roles
[Msg] [Dtf] End transaction on target server
[Msg] [Dtf] Get table data for messages
[Msg] [Dtf] Drop table: "members"
[Msg] [Dtf] Create table: "members"
[Msg] [Dtf] Begin transaction on target server
[Msg] [Dtf] Start transfer data for table: members
[Msg] [Dtf] End transaction on target server
[Msg] [Dtf] Drop table: "messages"
[Msg] [Dtf] Create table: "messages"
[Msg] [Dtf] Drop table: "milestone_versions"
[Msg] [Dtf] Get table data for milestone_versions
[Msg] [Dtf] Create table: "milestone_versions"

[Msg] [Dtf] Drop table: "milestones"
[Msg] [Dtf] Get table data for milestones
[Msg] [Dtf] Create table: "milestones"
[Msg] [Dtf] Drop table: "news"
[Msg] [Dtf] Get table data for news
[Msg] [Dtf] Create table: "news"
[Msg] [Dtf] Drop table: "next_issues"
[Msg] [Dtf] Get table data for next_issues
[Msg] [Dtf] Create table: "next_issues"
[Msg] [Dtf] Drop table: "notes"
[Msg] [Dtf] Get table data for notes
[Msg] [Dtf] Create table: "notes"
[Msg] [Dtf] Drop table: "open_id_authentication_associations"
[Msg] [Dtf] Get table data for open_id_authentication_associations
[Msg] [Dtf] Create table: "open_id_authentication_associations"
[Msg] [Dtf] Drop table: "open_id_authentication_nonces"
[Msg] [Dtf] Get table data for open_id_authentication_nonces
[Msg] [Dtf] Create table: "open_id_authentication_nonces"
[Msg] [Dtf] Drop table: "projects"
[Msg] [Dtf] Get table data for projects
[Msg] [Dtf] Create table: "projects"
[Msg] [Dtf] Begin transaction on target server
[Msg] [Dtf] Start transfer data for table: projects
[Msg] [Dtf] End transaction on target server
[Msg] [Dtf] Drop table: "projects_trackers"
[Msg] [Dtf] Get table data for projects_trackers
[Msg] [Dtf] Create table: "projects_trackers"
[Msg] [Dtf] Begin transaction on target server
[Msg] [Dtf] Start transfer data for table: projects_trackers
[Msg] [Dtf] End transaction on target server
[Msg] [Dtf] Drop table: "queries"
[Msg] [Dtf] Get table data for queries
[Msg] [Dtf] Create table: "queries"
[Msg] [Dtf] Begin transaction on target server
[Msg] [Dtf] Start transfer data for table: queries
[Msg] [Dtf] End transaction on target server
[Msg] [Dtf] Drop table: "recently_vieweds"
[Msg] [Dtf] Get table data for recently_vieweds
[Msg] [Dtf] Create table: "recently_vieweds"
[Msg] [Dtf] Drop table: "release_burndown_days"
[Msg] [Dtf] Get table data for release_burndown_days
[Msg] [Dtf] Create table: "release_burndown_days"
[Msg] [Dtf] Drop table: "releases"
[Msg] [Dtf] Get table data for releases
[Msg] [Dtf] Create table: "releases"
[Msg] [Dtf] Begin transaction on target server
[Msg] [Dtf] Start transfer data for table: releases
[Msg] [Dtf] End transaction on target server
[Msg] [Dtf] Drop table: "repositories"
[Msg] [Dtf] Get table data for repositories
[Msg] [Dtf] Create table: "repositories"
[Msg] [Dtf] Begin transaction on target server
[Msg] [Dtf] Start transfer data for table: repositories

[Msg] [Dtf] End transaction on target server
[Msg] [Dtf] Drop table: "roles"
[Msg] [Dtf] Get table data for roles
[Msg] [Dtf] Create table: "roles"
[Msg] [Dtf] Begin transaction on target server
[Msg] [Dtf] Start transfer data for table: roles
[Msg] [Dtf] End transaction on target server
[Msg] [Dtf] Drop table: "schema_migrations"
[Msg] [Dtf] Get table data for schema_migrations
[Msg] [Dtf] Create table: "schema_migrations"
[Msg] [Dtf] Begin transaction on target server
[Msg] [Dtf] Start transfer data for table: schema_migrations
[Msg] [Dtf] End transaction on target server
[Msg] [Dtf] Drop table: "settings"
[Msg] [Dtf] Get table data for settings
[Msg] [Dtf] Create table: "settings"
[Msg] [Dtf] Begin transaction on target server
[Msg] [Dtf] Start transfer data for table: settings
[Msg] [Dtf] End transaction on target server
[Msg] [Dtf] Drop table: "sprints_setups"
[Msg] [Dtf] Get table data for sprints_setups
[Msg] [Dtf] Create table: "sprints_setups"
[Msg] [Dtf] Drop table: "stuff_to_dos"
[Msg] [Dtf] Get table data for stuff_to_dos
[Msg] [Dtf] Create table: "stuff_to_dos"
[Msg] [Dtf] Begin transaction on target server
[Msg] [Dtf] Start transfer data for table: stuff_to_dos
[Msg] [Dtf] End transaction on target server
[Msg] [Dtf] Drop table: "taggings"
[Msg] [Dtf] Get table data for taggings
[Msg] [Dtf] Create table: "taggings"
[Msg] [Dtf] Drop table: "tags"
[Msg] [Dtf] Get table data for tags
[Msg] [Dtf] Create table: "tags"
[Msg] [Dtf] Drop table: "task_logs"
[Msg] [Dtf] Get table data for task_logs
[Msg] [Dtf] Create table: "task_logs"
[Msg] [Dtf] Drop table: "time_entries"
[Msg] [Dtf] Get table data for time_entries
[Msg] [Dtf] Create table: "time_entries"
[Msg] [Dtf] Begin transaction on target server
[Msg] [Dtf] Start transfer data for table: time_entries
[Msg] [Dtf] End transaction on target server
[Msg] [Dtf] Drop table: "time_estimates"
[Msg] [Dtf] Get table data for time_estimates
[Msg] [Dtf] Create table: "time_estimates"
[Msg] [Dtf] Drop table: "time_grid_issues_users"
[Msg] [Dtf] Get table data for time_grid_issues_users
[Msg] [Dtf] Create table: "time_grid_issues_users"
[Msg] [Dtf] Drop table: "tokens"
[Msg] [Dtf] Get table data for tokens
[Msg] [Dtf] Create table: "tokens"
[Msg] [Dtf] Begin transaction on target server

[Msg] [Dtf] Start transfer data for table: tokens
[Msg] [Dtf] End transaction on target server
[Msg] [Dtf] Drop table: "trackers"
[Msg] [Dtf] Get table data for trackers
[Msg] [Dtf] Create table: "trackers"
[Msg] [Dtf] Begin transaction on target server
[Msg] [Dtf] Start transfer data for table: trackers
[Msg] [Dtf] End transaction on target server
[Msg] [Dtf] Drop table: "user_preferences"
[Msg] [Dtf] Get table data for user_preferences
[Msg] [Dtf] Create table: "user_preferences"
[Msg] [Dtf] Begin transaction on target server
[Msg] [Dtf] Start transfer data for table: user_preferences
[Msg] [Dtf] End transaction on target server
[Msg] [Dtf] Drop table: "user_stories"
[Msg] [Dtf] Get table data for user_stories
[Msg] [Dtf] Create table: "user_stories"
[Msg] [Dtf] Begin transaction on target server
[Msg] [Dtf] Start transfer data for table: user_stories
[Msg] [Dtf] End transaction on target server
[Msg] [Dtf] Drop table: "users"
[Msg] [Dtf] Get table data for users
[Msg] [Dtf] Create table: "users"
[Msg] [Dtf] Begin transaction on target server
[Msg] [Dtf] Start transfer data for table: users
[Msg] [Dtf] End transaction on target server
[Msg] [Dtf] Drop table: "versions"
[Msg] [Dtf] Get table data for versions
[Msg] [Dtf] Create table: "versions"
[Msg] [Dtf] Begin transaction on target server
[Msg] [Dtf] Start transfer data for table: versions
[Msg] [Dtf] End transaction on target server
[Msg] [Dtf] Drop table: "watchers"
[Msg] [Dtf] Get table data for watchers
[Msg] [Dtf] Create table: "watchers"
[Msg] [Dtf] Begin transaction on target server
[Msg] [Dtf] Start transfer data for table: watchers
[Msg] [Dtf] End transaction on target server
[Msg] [Dtf] Drop table: "wiki_content_versions"
[Msg] [Dtf] Get table data for wiki_content_versions
[Msg] [Dtf] Create table: "wiki_content_versions"
[Msg] [Dtf] Begin transaction on target server
[Msg] [Dtf] Start transfer data for table: wiki_content_versions
[Msg] [Dtf] Get table data for wiki_contents
[Msg] [Dtf] End transaction on target server
[Msg] [Dtf] Drop table: "wiki_contents"
[Msg] [Dtf] Create table: "wiki_contents"
[Msg] [Dtf] Get table data for wiki_extensions_comments
[Msg] [Dtf] Begin transaction on target server
[Msg] [Dtf] Start transfer data for table: wiki_contents
[Msg] [Dtf] End transaction on target server
[Msg] [Dtf] Drop table: "wiki_extensions_comments"
[Msg] [Dtf] Create table: "wiki_extensions_comments"

[Msg] [Dtf] Drop table: "wiki_extensions_counts"
[Msg] [Dtf] Get table data for wiki_extensions_counts
[Msg] [Dtf] Create table: "wiki_extensions_counts"
[Msg] [Dtf] Begin transaction on target server
[Msg] [Dtf] Start transfer data for table: wiki_extensions_counts
[Msg] [Dtf] End transaction on target server
[Msg] [Dtf] Drop table: "wiki_extensions_menus"
[Msg] [Dtf] Get table data for wiki_extensions_menus
[Msg] [Dtf] Create table: "wiki_extensions_menus"
[Msg] [Dtf] Begin transaction on target server
[Msg] [Dtf] Start transfer data for table: wiki_extensions_menus
[Msg] [Dtf] End transaction on target server
[Msg] [Dtf] Drop table: "wiki_extensions_settings"
[Msg] [Dtf] Get table data for wiki_extensions_settings
[Msg] [Dtf] Create table: "wiki_extensions_settings"
[Msg] [Dtf] Begin transaction on target server
[Msg] [Dtf] Start transfer data for table: wiki_extensions_settings
[Msg] [Dtf] End transaction on target server
[Msg] [Dtf] Drop table: "wiki_extensions_tag_relations"
[Msg] [Dtf] Get table data for wiki_extensions_tag_relations
[Msg] [Dtf] Create table: "wiki_extensions_tag_relations"
[Msg] [Dtf] Drop table: "wiki_extensions_tags"
[Msg] [Dtf] Get table data for wiki_extensions_tags
[Msg] [Dtf] Create table: "wiki_extensions_tags"
[Msg] [Dtf] Drop table: "wiki_extensions_votes"
[Msg] [Dtf] Get table data for wiki_extensions_votes
[Msg] [Dtf] Create table: "wiki_extensions_votes"
[Msg] [Dtf] Drop table: "wiki_pages"
[Msg] [Dtf] Get table data for wiki_pages
[Msg] [Dtf] Create table: "wiki_pages"
[Msg] [Dtf] Begin transaction on target server
[Msg] [Dtf] Start transfer data for table: wiki_pages
[Msg] [Dtf] End transaction on target server
[Msg] [Dtf] Drop table: "wiki_redirects"
[Msg] [Dtf] Get table data for wiki_redirects
[Msg] [Dtf] Create table: "wiki_redirects"
[Msg] [Dtf] Begin transaction on target server
[Msg] [Dtf] Start transfer data for table: wiki_redirects
[Msg] [Dtf] End transaction on target server
[Msg] [Dtf] Drop table: "wikis"
[Msg] [Dtf] Get table data for wikis
[Msg] [Dtf] Create table: "wikis"
[Msg] [Dtf] Begin transaction on target server
[Msg] [Dtf] Start transfer data for table: wikis
[Msg] [Dtf] End transaction on target server
[Msg] [Dtf] Drop table: "workflows"
[Msg] [Dtf] Get table data for workflows
[Msg] [Dtf] Create table: "workflows"
[Msg] [Dtf] Begin transaction on target server
[Msg] [Dtf] Start transfer data for table: workflows
[Msg] [Dtf] End transaction on target server
[Msg] [Dtf] Unlock source server tables
[Msg] [Dtf] Finished successfully

I however still get the same error:

Started POST "/login" for 217.164.33.239 at 2012-09-22 08:39:05 +0000
Processing by AccountController#login as HTML
Parameters: {"utf8"=>"", "authenticity_token"=>"38gqDOX+4/Yn35t+gRjMSSGTAWXNygsSX2S5SCWYUhk=", "back_url"=>"http://red.nuserv.com/", "username"=>"khelal", "password"=>"[FILTERED]", "login"=>"Login »"}
Current user: anonymous
Successful authentication for 'khelal' from 217.164.33.239 at 2012-09-22 08:39:05 UTC
Redirected to http://red.nuserv.com/
Completed 302 Found in 258ms (ActiveRecord: 12.3ms)

Started GET "/" for 217.164.33.239 at 2012-09-22 08:39:05 +0000
Processing by WelcomeController#index as HTML
Current user: khelal (id=1)
Rendered welcome/index.html.erb within layouts/base (45.3ms)
Completed 500 Internal Server Error in 127ms

ActionView::Template::Error (can't convert Symbol into Integer):

```
9: <%= favicon %>
10: <%= stylesheet_link_tag 'jquery/jquery-ui-1.8.21', 'application', :media => 'all' %>
11: <%= stylesheet_link_tag 'rtl', :media => 'all' if l(:direction) == 'rtl' %>
12: <%= javascript_heads %>
13: <%= heads_for_theme %>
14: <%= call_hook :view_layouts_base_html_head %>
15: <!-- page specific tags -->
app/models/user_preference.rb:39:in `[]'
app/models/user_preference.rb:39:in `[]'
app/models/user_preference.rb:57:in `warn_on_leaving_unsaved'
app/helpers/application_helper.rb:1222:in `javascript_heads'
app/views/layouts/base.html.erb:12:in `_app_views_layouts_base_html_erb__3650353151514419733_39746260'
```

#5 - 2012-09-22 11:15 - Jean-Philippe Lang

Still, something is wrong with your new database. A safer process would be to create a blank postgres database then create the structure using db:migrate then import only the data from your old database. To solve the last error you've posted, you can try to reset user's preference: UPDATE user_preferences SET others = NULL; But this migration from mysql to postgresql is out of the scope of this ticket.

#6 - 2012-09-25 17:20 - Etienne Massip

- Category set to Code cleanup/refactoring

#7 - 2012-09-29 18:33 - Jean-Philippe Lang

- Subject changed from can't dup NilClass error after upgrading to 2.1.0 to Potential "can't dup NilClass" error in UserPreference
- Status changed from Resolved to Closed