

## Redmine - Defect #35823

### undefined method `+' for nil:NilClass

2021-08-28 09:16 - Martin K

<b>Status:</b>	Closed	<b>Start date:</b>	
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Go MAEDA	<b>% Done:</b>	0%
<b>Category:</b>	Issues	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>		<b>Affected version:</b>	4.2.2
<b>Resolution:</b>	Fixed		

#### Description

Hi,

I am using latest Redmine SVN version, I am having trouble opening some issues (some are OK, some cause error). This is what I get:

```
Started GET "/issues/4617" for xx.xx.xx.xx at 2021-08-28 09:07:21 +0200
```

```
Processing by IssuesController#show as HTML
```

```
Parameters: {"id"=>"4617"}
```

```
Current user: xxxx (id=3)
```

```
Rendered issues/show.html.erb within layouts/base (Duration: 996.0ms | Allocations: 133997)
```

```
Rendered layout layouts/base.html.erb (Duration: 996.6ms | Allocations: 134036)
```

```
Completed 500 Internal Server Error in 2302ms (ActiveRecord: 371.0ms | Allocations: 325712)
```

```
ActionView::Template::Error (undefined method `+' for nil:NilClass):
```

```
10:
```

```
11: <h3><%= I(:"label_#{watched_klass_name}_watchers") %> (<%= watched.watcher_users.size %>)</h3>
```

```
12:
```

```
13: <%= watchers_list(watched) %>
```

```
app/helpers/application_helper.rb:75:in `link_to_principal'
```

```
app/helpers/application_helper.rb:54:in `link_to_user'
```

```
app/helpers/watchers_helper.rb:54:in `block in watchers_list'
```

```
app/helpers/watchers_helper.rb:51:in `collect'
```

```
app/helpers/watchers_helper.rb:51:in `watchers_list'
```

```
app/views/watchers/_watchers.html.erb:13
```

```
app/views/issues/show.html.erb:142
```

```
app/views/issues/show.html.erb:136
```

```
app/controllers/issues_controller.rb:118:in `block (2 levels) in show'
```

```
app/controllers/issues_controller.rb:110:in `show'
```

```
lib/redmine/sudo_mode.rb:61:in `sudo_mode'
```

```
$ruby bin/about
```

```
sh: hg: command not found
```

```
sh: cvs: command not found
```

```
sh: bzip2: command not found
```

```
Environment:
```

```
Redmine version      4.2.2.devel.21201
```

```
Ruby version         2.5.9-p229 (2021-04-05) [x86_64-linux]
```

```
Rails version        6.1.4.1
```

```
Environment          production
```

Database adapter	PostgreSQL
Mailer queue	ActiveJob::QueueAdapters::AsyncAdapter
Mailer delivery	smtp

SCM:

Subversion	1.9.7
Git	2.14.3

Filesystem

Redmine plugins:

periodictask	4.1.0
projects_table	0.0.4
redmine_editauthor	0.11.0
redmine_impersonate	0.10.0
redmine_timelog_timer	2.0.1
sidebar_hide	0.0.8

BTW: also happens if all plugins are disabled

#### Related issues:

Related to Redmine - Feature # 12795: View group members by non-admin users

Closed

#### Associated revisions

##### Revision 21217 - 2021-09-07 16:59 - Go MAEDA

Fix NoMethodError when generating a link to a locked user (#12795, #35823).

#### History

##### #1 - 2021-08-28 09:24 - Go MAEDA

Could you test if changing app/helpers/application\_helper.rb as follows fixes the error?

```
diff --git a/app/helpers/application_helper.rb b/app/helpers/application_helper.rb
index e3c858db4..b97b2fc98 100644
--- a/app/helpers/application_helper.rb
+++ b/app/helpers/application_helper.rb
@@ -70,6 +70,7 @@ module ApplicationHelper
   css_classes = 'group'
   else
     name = h(principal.to_s)
+   css_classes = "
   end

   css_classes += " #{options[:class]}" if options[:class].present?
```

##### #2 - 2021-08-28 09:24 - Go MAEDA

- Related to Feature #12795: View group members by non-admin users added

##### #3 - 2021-08-28 10:23 - Martin K

Sorry, still same error

Martin K wrote:

| *Sorry, still same error*

I think I understand why. Please revert that change and try the following:

```
app/helpers/application_helper.rb | 3 ++-
```

```
1 file changed, 2 insertions(+), 1 deletion(-)
```

```
diff --git a/app/helpers/application_helper.rb b/app/helpers/application_helper.rb
index e3c858db4..8f4ea5255 100644
--- a/app/helpers/application_helper.rb
+++ b/app/helpers/application_helper.rb
@@ -60,9 +60,10 @@ module ApplicationHelper
  case principal
  when User
    name = h(principal.name(options[:format]))
+   css_classes = "
    if principal.active? || (User.current.admin? && principal.logged?)
      url = user_url(principal, :only_path => only_path)
-   css_classes = principal.css_classes
+   css_classes += principal.css_classes
  end
  when Group
    name = h(principal.to_s)
```

@Go: this error is produced while rendering the issue user watchers on the issues show view. This is, besides the explicitly given code snippet, given by these lines in the stack trace:

```
app/helpers/watchers_helper.rb:54:in `block in watchers_list'
app/helpers/watchers_helper.rb:51:in `collect'
app/helpers/watchers_helper.rb:51:in `watchers_list'
app/views/watchers/_watchers.html.erb:13
app/views/issues/show.html.erb:142
app/views/issues/show.html.erb:136
```

Now if we go back to the error that is given, `ActionView::Template::Error (undefined method `+' for nil:NilClass)`, we see that Redmine tries to send the `+` message to a nil object. The caller in that case is given by the first line (from the top) of the stack trace:

```
app/helpers/application_helper.rb:75:in `link_to_principal'
```

So, in the cases the error occurs `css_classes` is nil. Given that we're in the method `link_to_principal` and we're trying to send `+` to an object, I infer that `css_classes` should have been set<sup>1</sup> prior to sending it this message. I look at the method and see that `css_classes` is set in several of the `when` branches of the case statement. We see that the case statement is passed the `principal` variable, so we need to figure out what branch we're on when

the error occurs.

Now the value of principal is determined by what is passed along when link\_to\_principal was called. So we need to know what called link\_to\_principal when and with what arguments, thus we go back to the call stack and observe that it reads on the second line:

```
app/helpers/application_helper.rb:54:in `link_to_user'
```

Now, link\_to\_user has two execution paths based on the condition user.is\_a?(User). link\_to\_user exits with the call to link\_to\_principal so we now know that the error occurs on a User object.

We then go back to the when User branch of the case statement in link\_to\_principal and determine that in the case that the error occurs, the assignment css\_classes = principal.css\_classes doesn't work as expected. Now, when we look at this code we see that this assignment is reached conditionally via the if-statement that it's wrapped in. The condition being: "principal.active? || (User.current.admin? && principal.logged?)".

We now have reached the root cause of the issue: if !principal.active? or !(User.current.admin? && principal.logged?) we don't assign css\_classes with principal.css\_classes, hence is it nil by the time we send + to it on line 75.

This can as such be fixed by assigning css\_classes with a blank string prior to the if-statement, and then subsequently add to that blank string (by sending it +=) the return value of principal.css\_classes inside the if-statement.

Note: this error would also occur if the principal is something other than a User or Group object and link\_to\_principal is called and passed the :class option, which would then indeed be fixed by the patch you propose in note#1. However, I think that the CSS classes should be provided by the actual subclasses of Principal. They should implement a css\_classes method to return this information as part of the model. As such:

- should css\_classes = 'group' be refactored into an assignment with the return value of a to-be-created Group#css\_classes method
- should be determined if link\_to\_principal should handle unknown Principals. Given that link\_to\_user prior to r21073 "handled" this by just returning a string representation of the principal, no class was needed. If link\_to\_principal is modeled after link\_to\_user prior to r21073, as it currently is, note#1 patch is not needed. However, in that case should the css\_classes += "#{options[:class]}" if options[:class].present? assignment be done in both the User and Group branches of the case-statement.

Please let me know if more information is needed.

<sup>1</sup> Be aware of the scope of local variables within case statements here also; i.e. case along with if, unless, for and while *doesn't create a new scope* in Ruby. See e.g. [this](#) question on StackOverflow and the references given therein.

#### #5 - 2021-08-29 04:34 - Mischa The Evil

- Description updated

#### #6 - 2021-08-29 11:48 - Martin K

That helped, thanks! :)

#### #7 - 2021-08-31 04:43 - Go MAEDA

Thanks to the detailed explanation #35823#note-4 posted by Mischa, I understand what is causing the issue.

The following test catches the reported issue.

```
diff --git a/test/helpers/application_helper_test.rb b/test/helpers/application_helper_test.rb
index 06dd04b6a..2d7dd0fc5 100644
--- a/test/helpers/application_helper_test.rb
+++ b/test/helpers/application_helper_test.rb
@@ -1701,12 +1701,18 @@ class ApplicationHelperTest < Redmine::HelperTest
  end
```

end

```
- def test_link_to_user_should_link_to_locked_user_if_current_user_is_admin
+ def test_link_to_user_should_link_to_locked_user_only_if_current_user_is_admin
+   user = User.find(5)
+   assert user.locked?
+
+   with_current_user User.find(1) do
-     user = User.find(5)
-     assert user.locked?
-     result = link_to("Dave2 Lopper2", "/users/5", :class => "user locked")
-     assert_equal result, link_to_user(user)
+     result = link_to("Dave2 Lopper2", '/users/5', :class => 'user locked assigned_to')
+     assert_equal result, link_to_user(user, :class => 'assigned_to')
+   end
+
+   with_current_user User.find(2) do
+     result = 'Dave2 Lopper2'
+     assert_equal result, link_to_user(user, :class => 'assigned_to')
  end
end
```

#### #8 - 2021-09-04 10:24 - Go MAEDA

- File 35823.patch added

Here is a patch to fix this issue:

- includes the fix #35823#note-4
- includes the test #35823#note-7
- implements Group#css\_classes and link\_to\_principal uses the method
- link\_to\_principal just returns a string representation for an unknown type principal. The behavior is the same as before r21073

#### #9 - 2021-09-07 07:53 - Martin K

should I set it to resolved or will somebody do it when it gets to SVN?

#### #10 - 2021-09-07 17:00 - Go MAEDA

- Status changed from New to Closed
- Assignee set to Go MAEDA
- Resolution set to Fixed

Committed the fix in r21217 as a part of #12795.

#### Files

---

35823.patch	2.9 KB	2021-09-04	Go MAEDA
-------------	--------	------------	----------