Redmine - Defect #11701

Cannot use RSS-feed after issues importing

2012-08-24 17:42 - # And

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			
Resolution:	Invalid	**Affected version:**	2.0.3

Description

Hi.

I have used plugin to import csv file, and update existent issues. After this, when I go to rss-feed or open my user page I got 500 error. Here is log saying:

```
Started GET "/projects/btt/issues.atom?key=e736bac674718e1291974915ecf0379d89cce4d3" for 46.211.120.252 at 2012-08-24 17:25:16 +0200
Processing by IssuesController#index as ATOM
  Parameters: {"key"=>"e736bac674718e1291974915ecf0379d89cce4d3", "project_id"=>"btt"}
Completed 500 Internal Server Error in 117ms

NoMethodError (undefined method `to_datetime' for nil:NilClass):
  app/controllers/application_controller.rb:345:in `block in render_feed'
  app/controllers/application_controller.rb:345:in `sort!'
  app/controllers/application_controller.rb:345:in `render_feed'
  app/controllers/issues_controller.rb:87:in `block (2 levels) in index'
  app/controllers/issues_controller.rb:82:in `index'
```

or user page error:

```
Started GET "/users/4" for 46.211.120.252 at 2012-08-24 17:39:39 +0200
Processing by UsersController#show as HTML
  Parameters: {"id"=>"4"}
Completed 500 Internal Server Error in 99ms

NoMethodError (undefined method `to_datetime' for nil:NilClass):
  lib/redmine/activity/fetcher.rb:80:in `block in events'
  lib/redmine/activity/fetcher.rb:80:in `sort!'
  lib/redmine/activity/fetcher.rb:80:in `events'
  app/controllers/users_controller.rb:68:in `show'
```

How can I fix it? Or it is a redmine bug (no checking for the variable type)? Maybe importer not write properly change date of the activity?

Thanks.

History

#1 - 2012-08-25 22:50 - # And

- Status changed from New to Resolved

I tried to resolve this issue by itself, and what I found:

```
def events(from = nil, to = nil, options={})
  e = []
  @options[:limit] = options[:limit]

  @scope.each do |event_type|
    constantized_providers(event_type).each do |provider|
      e += provider.find_events(event_type, @user, from, to, @options)
    end
  end

  e.sort! {|a,b| b.event_datetime <=> a.event_datetime}

  if options[:limit]
    e = e.slice(0, options[:limit])
  end
  e
end
```

What is **e += provider.find_events(event_type, user user, from, to, @options)** doing? What is the **event**?

added:
After debugging I found issue without **.event_datetime**. It is issue without **created_on** value in the issues table. I fill this column, and it works.

Cool! =)

#2 - 2012-08-26 12:20 - Etienne Massip

- Status changed from Resolved to Closed

- Resolution set to Invalid

Thanks.