

Redmine - Defect #969

Custom Field causes 500 internal server error when adding tracker items

2008-04-01 01:34 - Robert Prince

Status:	Closed	Start date:	2008-04-01
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:	Custom fields	Estimated time:	0.00 hour
Target version:	0.7	Affected version:	
Resolution:	Fixed		
Description			
<p>I added a custom field and applied it to a single tracker (I have multiple trackers in my redmine installation). It was a text field, no min/max, regex, or default value. When I subsequently added items to that tracker, the system would throw a '500 internal server error'. When I looked in the log to see what was going on, and saw where in the issues_controller it was happening, I realized it might be the custom field, so I removed it, and now the tracker works again.</p> <p>FYI, not sure it matters but I'm running redmine under lighttpd/fastcgi.</p> <p>I grabbed the following from the log while doing some troubleshooting:</p> <pre>Processing IssuesController#new (for 72.215.161.251 at 2008-03-31 19:15:42) [POST] Session ID: 549d77609f2e2ca19099cd76a51b3749 Parameters: {"project_id"=>"ls-fwk-5", "commit"=>"Create", "action"=>"new", "controller"=>"issues", "issue"=>{"start_date"=>"2008-03-31", "estimated_hours"=>""}, "priority_id"=>"3", "subject"=>"another test", "description"=>"test ignore", "tracker_id"=>"10", "fixed_version_id"=>""}, "category_id"=>""}, "done_ratio"=>"0", "due_date"=>""}, "assigned_to_id"=>""}, "status_id"=>"1"}, "attachments"=>{"1"=>{"description"=>""}, "file"=>""}} NoMethodError (undefined method `[]' for nil:NilClass): /app/controllers/issues_controller.rb:143:in `new' /app/controllers/issues_controller.rb:143:in `collect' /app/controllers/issues_controller.rb:143:in `new' /usr/lib/ruby/gems/1.8/gems/actionpack-1.13.3/lib/action_controller/base.rb:1095:in `send' /usr/lib/ruby/gems/1.8/gems/actionpack-1.13.3/lib/action_controller/base.rb:1095:in `perform_action_without_filters' /usr/lib/ruby/gems/1.8/gems/actionpack-1.13.3/lib/action_controller/filters.rb:632:in `call_filter' /usr/lib/ruby/gems/1.8/gems/actionpack-1.13.3/lib/action_controller/filters.rb:634:in `call_filter' /usr/lib/ruby/gems/1.8/gems/actionpack-1.13.3/lib/action_controller/filters.rb:638:in `call_filter' /usr/lib/ruby/gems/1.8/gems/actionpack-1.13.3/lib/action_controller/filters.rb:438:in `call' /usr/lib/ruby/gems/1.8/gems/actionpack-1.13.3/lib/action_controller/filters.rb:637:in `call_filter' /usr/lib/ruby/gems/1.8/gems/actionpack-1.13.3/lib/action_controller/filters.rb:638:in `call_filter' /usr/lib/ruby/gems/1.8/gems/actionpack-1.13.3/lib/action_controller/filters.rb:438:in `call' /usr/lib/ruby/gems/1.8/gems/actionpack-1.13.3/lib/action_controller/filters.rb:637:in `call_filter' /usr/lib/ruby/gems/1.8/gems/actionpack-1.13.3/lib/action_controller/filters.rb:634:in `call_filter' /usr/lib/ruby/gems/1.8/gems/actionpack-1.13.3/lib/action_controller/filters.rb:638:in `call_filter' /usr/lib/ruby/gems/1.8/gems/actionpack-1.13.3/lib/action_controller/filters.rb:438:in `call' /usr/lib/ruby/gems/1.8/gems/actionpack-1.13.3/lib/action_controller/filters.rb:637:in `call_filter' /usr/lib/ruby/gems/1.8/gems/actionpack-1.13.3/lib/action_controller/filters.rb:638:in `call_filter' /usr/lib/ruby/gems/1.8/gems/actionpack-1.13.3/lib/action_controller/filters.rb:438:in `call' /usr/lib/ruby/gems/1.8/gems/actionpack-1.13.3/lib/action_controller/filters.rb:637:in `call_filter' /usr/lib/ruby/gems/1.8/gems/actionpack-1.13.3/lib/action_controller/filters.rb:638:in `call_filter' /usr/lib/ruby/gems/1.8/gems/actionpack-1.13.3/lib/action_controller/filters.rb:438:in `call' /usr/lib/ruby/gems/1.8/gems/actionpack-1.13.3/lib/action_controller/filters.rb:637:in `call_filter' /usr/lib/ruby/gems/1.8/gems/actionpack-1.13.3/lib/action_controller/filters.rb:638:in `call_filter' /usr/lib/ruby/gems/1.8/gems/actionpack-1.13.3/lib/action_controller/filters.rb:619:in `perform_action_without_benchmark'</pre>			

```
/usr/lib/ruby/gems/1.8/gems/actionpack-1.13.3/lib/action_controller/benchmarking.rb:66:in `perform_action_without_rescue'  
/usr/lib/ruby/1.8/benchmark.rb:293:in `measure'  
/usr/lib/ruby/gems/1.8/gems/actionpack-1.13.3/lib/action_controller/benchmarking.rb:66:in `perform_action_without_rescue'  
/usr/lib/ruby/gems/1.8/gems/actionpack-1.13.3/lib/action_controller/rescue.rb:83:in `perform_action'  
/usr/lib/ruby/gems/1.8/gems/actionpack-1.13.3/lib/action_controller/base.rb:430:in `send'  
/usr/lib/ruby/gems/1.8/gems/actionpack-1.13.3/lib/action_controller/base.rb:430:in `process_without_filters'  
/usr/lib/ruby/gems/1.8/gems/actionpack-1.13.3/lib/action_controller/filters.rb:624:in  
`process_without_session_management_support'  
/usr/lib/ruby/gems/1.8/gems/actionpack-1.13.3/lib/action_controller/session_management.rb:114:in `process'  
/usr/lib/ruby/gems/1.8/gems/actionpack-1.13.3/lib/action_controller/base.rb:330:in `process'  
/usr/lib/ruby/gems/1.8/gems/rails-1.2.3/lib/dispatcher.rb:41:in `dispatch'  
/usr/lib/ruby/gems/1.8/gems/rails-1.2.3/lib/cgi_handler.rb:168:in `process_request'  
/usr/lib/ruby/gems/1.8/gems/rails-1.2.3/lib/cgi_handler.rb:143:in `process_each_request!'  
/usr/lib/ruby/gems/1.8/gems/rails-1.2.3/lib/cgi_handler.rb:109:in `with_signal_handler'  
/usr/lib/ruby/gems/1.8/gems/rails-1.2.3/lib/cgi_handler.rb:142:in `process_each_request!'  
/usr/lib/ruby/site_ruby/1.8/cgi.rb:600:in `each CGI'  
/usr/lib/ruby/site_ruby/1.8/cgi.rb:117:in `session'  
/usr/lib/ruby/site_ruby/1.8/cgi.rb:104:in `each_request'  
/usr/lib/ruby/site_ruby/1.8/cgi.rb:36:in `each'  
/usr/lib/ruby/site_ruby/1.8/cgi.rb:597:in `each CGI'  
/usr/lib/ruby/gems/1.8/gems/rails-1.2.3/lib/cgi_handler.rb:141:in `process_each_request!'  
/usr/lib/ruby/gems/1.8/gems/rails-1.2.3/lib/cgi_handler.rb:55:in `process!'  
/usr/lib/ruby/gems/1.8/gems/rails-1.2.3/lib/cgi_handler.rb:25:in `process!'  
/mnt/redmine/public/dispatch.fcgi:27
```

Associated revisions

Revision 1317 - 2008-04-01 19:48 - Jean-Philippe Lang

Prevent NoMethodError on nil class if custom_fields params is not present in IssuesController#new (#969).

History

#1 - 2008-04-01 19:54 - Jean-Philippe Lang

- Category set to *Custom fields*
- Status changed from *New* to *Resolved*
- Target version set to *0.7*
- Resolution set to *Fixed*

I've committed a fix in trunk (r1317) to prevent this error but it shouldn't occur if the custom fields were properly submitted when creating the issue.

I have a few questions:

- Do you see the custom field when filling the issue form ?
- Is the issue form updated when you change the tracker on the issue form (an ajax request should happen) ?
- Is your custom field specific to a tracker ?

#2 - 2008-04-27 17:48 - Jean-Philippe Lang

- Status changed from *Resolved* to *Closed*

Upgrading to rails 1.2.6 will solve this problem.

