

Redmine - Defect #6403

Updating an issue with custom fields fails

2010-09-15 18:13 - Stéphane Dubois

Status:	Closed	Start date:	2010-09-15
Priority:	Normal	Due date:	
Assignee:	Jean-Philippe Lang	% Done:	0%
Category:	REST API	Estimated time:	0.00 hour
Target version:	1.1.0	Affected version:	1.0.4
Resolution:	Fixed		

Description

If I run the following code

```
require 'rubygems'
require 'active_resource'

# Issue model on the client side
class Issue < ActiveResource::Base
  self.site = 'http://redmine.server/'
  self.user = 'foo'
  self.password = 'bar'
end

# Updating an issue
issue = Issue.find(1)
issue.subject = 'REST API'
issue.save
```

it fails on issue.save with the following error

```
C:/Ruby187/lib/ruby/gems/1.8/gems/activesupport-2.3.5/lib/active_support/core_ext/array/conversions.rb:163:in `to_xml': Not all elements respond to to_xml (RuntimeError)
```

I suspect it has something to do with custom fields because if I do

```
# Updating an issue
issue = Issue.find(1)
issue.subject = 'REST API'
issue.custom_fields = nil
issue.save
```

it works !

Related issues:

Related to Redmine - Defect #7377: Updating an Issue with custom fields set f...	Closed	2011-01-19
--	--------	------------

Associated revisions

Revision 4480 - 2010-12-10 11:12 - Jean-Philippe Lang

Fixed: error when serializing back objects with custom fields using ActiveResource (#6403).

Revision 4481 - 2010-12-10 11:48 - Jean-Philippe Lang

Support for updating custom fields using the received custom_fields array (#6345, #6403).

History

#1 - 2010-09-16 23:25 - Ward V

I can confirm this - I'm seeing the same thing. It's the custom_fields that cause this.

#2 - 2010-09-17 15:44 - Stéphane Dubois

Something strange:
to create an issue with a custom field we do

```
issue = Issue.new(  
  :subject => 'REST API',  
  :assigned_to_id => 1,  
  :project_id => 1,  
  :custom_field_values => {'2' => 'Fixed'}  
)  
issue.save
```

but when I retrieve an issue with REST API, `:custom_field_values` is not an entry of the attributes hash. Instead, I have a `:custom_fields` entry containing a `custom_field` entry which is an array containing all custom fields values.
If a try to update one of those values like that

```
issue = Issue.find(1)  
issue.custom_fields.custom_field[0] = dummy_value  
issue.save
```

I get the same error

```
C:/Ruby187/lib/ruby/gems/1.8/gems/activesupport-2.3.5/lib/active_support/core_ext/array/conversions.rb:163:in  
`to_xml': Not all elements respond to to_xml (RuntimeError)
```

So, I can't find a way to update a custom field.

#3 - 2010-09-18 00:02 - Stéphane Dubois

In fact, it's possible to update a custom fields with the same hash than for creation, but you must also nilify `custom_fields`

```
issue = Issue.find(1)  
issue.custom_field_values = {'1' => 'dummy'}  
issue.custom_fields = nil  
issue.save
```

#4 - 2010-11-01 16:04 - Eric Davis

- Assignee set to Eric Davis

#5 - 2010-11-27 00:40 - Eric Davis

- Assignee deleted (Eric Davis)

I am stepping down from working on Redmine. If someone else is interesting in working on this issue, feel free to reassign it to them.

Eric Davis

#6 - 2010-12-03 19:37 - Jean-Philippe Lang

Stéphane Dubois wrote:

```
In fact, it's possible to update a custom fields with the same hash than for creation, but you must also nilify custom_fields  
[...]
```

I don't know which version you're using but I can't reproduce this problem with current trunk. The following works:

```
issue = Issue.find(1)  
issue.custom_field_values = {'1' => 'dummy'}  
issue.save
```

#7 - 2010-12-08 20:52 - Jean-Philippe Lang

- Assignee set to Jean-Philippe Lang

- Target version set to 1.0.5

- Affected version (unused) set to 1.0.4

- Affected version set to 1.0.4

Problem found. The fix requires a slight change in the custom fields part of the API.

#8 - 2010-12-10 11:48 - Jean-Philippe Lang

- *Status changed from New to Resolved*

- *Resolution set to Fixed*

Problem is fixed in [r4480](#), [r4481](#). You can now use the custom_fields array received in the API response to update custom_fields.

```
issue = Issue.find(6467)
issue.custom_fields.first.name # => "Affected version"
issue.custom_fields.first.value = '1.0.3'
issue.save
```

#9 - 2010-12-10 12:33 - Jean-Philippe Lang

- *Status changed from Resolved to Closed*

- *Target version changed from 1.0.5 to 1.1.0*

It will have to wait 1.1.0 because it relies on changes done in trunk only.