

Redmine - Defect #7834

API Issue XML cannot represent empty arrays

2011-03-11 11:15 - milki milk

Status:	New	Start date:	2011-03-11
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:	REST API	Estimated time:	0.00 hour
Target version:		Affected version:	
Resolution:			

Description

When trying to retrieve issues from a project with no issues, there is a typecast issue from the xml conversion.

I am using Redmine trunk, Rails 2.3.11

Sample code:

```
#!/usr/local/bin/ruby

require 'rubygems'
require 'active_resource'

class Issue < ActiveResource::Base
  self.site = 'http://demo.redmine.org/'
end

issue = Issue.find(:first, :params => { :project_id => 3 })
```

And the output:

```
/usr/local/lib/ruby/gems/1.8/gems/activesupport-2.3.11/lib/active_support/core_ext/hash/conversions.rb:99:in `typecast_xml_value': can't typecast "" (RuntimeError)
    from /usr/local/lib/ruby/gems/1.8/gems/activesupport-2.3.11/lib/active_support/core_ext/hash/conversions.rb:121:in `typecast_xml_value'
    from /usr/local/lib/ruby/site_ruby/1.8/rubygems/custom_require.rb:29:in `inject'
    from /usr/local/lib/ruby/gems/1.8/gems/activesupport-2.3.11/lib/active_support/core_ext/hash/conversions.rb:120:in `each'
    from /usr/local/lib/ruby/gems/1.8/gems/activesupport-2.3.11/lib/active_support/core_ext/hash/conversions.rb:120:in `inject'
    from /usr/local/lib/ruby/gems/1.8/gems/activesupport-2.3.11/lib/active_support/core_ext/hash/conversions.rb:120:in `typecast_xml_value'
    from /usr/local/lib/ruby/gems/1.8/gems/activesupport-2.3.11/lib/active_support/core_ext/hash/conversions.rb:78:in `from_xml'
    from /usr/local/lib/ruby/gems/1.8/gems/activeresource-2.3.11/lib/active_resource/formats/xml_format.rb:22:in `decode'
    from /usr/local/lib/ruby/gems/1.8/gems/activeresource-2.3.11/lib/active_resource/connection.rb:79:in `get'
    from /usr/local/lib/ruby/gems/1.8/gems/activeresource-2.3.11/lib/active_resource/connection.rb:217:in `with_auth'
    from /usr/local/lib/ruby/gems/1.8/gems/activeresource-2.3.11/lib/active_resource/connection.rb:79:in `get'
    from /usr/local/lib/ruby/gems/1.8/gems/activeresource-2.3.11/lib/active_resource/base.rb:857:in `find_every'
    from /usr/local/lib/ruby/gems/1.8/gems/activeresource-2.3.11/lib/active_resource/base.rb:778:in `find'
```

The problem is different from [#4745](#) and [#5226](#). Let's examine the xml returned from the call:

```
<?xml version="1.0" encoding="UTF-8"?><issues type="array" offset="0" total_count="0" limit="25"></issues>
```

Ok, that looks fine. Let's see how it's turned into an ActiveResource:

```
{"type"=>"array", "total_count"=>"0", "limit"=>"25", "offset"=>"0"}
```

This last line is passed into `typecast_xml_value` in `/usr/local/lib/ruby/gems/1.8/gems/activesupport-2.3.11/lib/active_support/core_ext/hash/conversions.rb`. In this code, it will do the following:

```
_, entries = Array.wrap(value.detect { |k,v| k != 'type' })
if entries.nil? || (c = value['__content__'] && c.blank?)
  []
else
  case entries.class.to_s
  when "Array"
  when "Hash"
  else
    raise "can't typecast #{entries.inspect}"
  end
end
```

The first line sets `entries` to `"0"` (or whatever is the value of the next key), so it will always hit the exception. The second line shows that an explicit empty key `content` is used to detect an empty array. Thus, if I patch `index.api.rsb`:

```
+if @issues.empty?
+  api.array :issues, api_meta( :__content__ => '', :total_count => @issue_count, :offset => @offset, :limit => @limit)
+else
+  ...
end
```

Then the returned xml is correct. This patch will not work however because it interferes with the json format, which works fine by representing an empty array with `[]`. I've attached the simple script, a proposed patch that fixes the problem but breaks json, and a test. It may be that the json/xml template needs to be split again.

Files			
patch-index.api.rsb	3.32 KB	2011-03-11	milki milk
patch-issues_test.rb	679 Bytes	2011-03-11	milki milk
fail.rb	227 Bytes	2011-03-11	milki milk