

Redmine - Patch #21841

Correctly export collections associated with issues.

2016-01-25 17:32 - Anonymous

Status:	Closed	Start date:	
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:	Issues	Estimated time:	0.00 hour
Target version:			
Description			
When you have a plugins that add a 1:m association to issues (say tags) and you want to export the associated entities the CSV and PDF exports give you a ActiveRecord_Associations_CollectionProxy object serialization as the column value.			
This patch Fixes the behaviour for all mappable (or collectable if you prefer) objects (see attached before and after exports).			

History

#1 - 2016-02-22 06:14 - Toshi MARUYAMA

How about you use #to_a?

http://api.rubyonrails.org/classes/ActiveRecord/Associations/CollectionProxy.html#method-i-to_a

#2 - 2016-02-22 11:20 - Anonymous

Toshi MARUYAMA wrote:

How about you use #to_a?

http://api.rubyonrails.org/classes/ActiveRecord/Associations/CollectionProxy.html#method-i-to_a

Where exactly do you suggest I use the **to_a** method?

Both exports (CSV and PDF) loop trough the available columns of the issue query and format the values for display. There is no place to specify that a certain value is a collection or to inject a transformation.

Rails uses **to_a** when calling a method from Enumerable (ex: collect) on an AR collection. This patch ensures that it gets called when the appropriate type is given (it could be a plain array or a proxy that resolves to an array).

#3 - 2016-03-23 05:07 - Toshi MARUYAMA

Codruț Gușoi wrote:

Where exactly do you suggest I use the **to_a** method?

For example: [r13248](#).

#4 - 2016-03-23 05:10 - Toshi MARUYAMA

In your patch:

```
diff --git a/lib/redmine/export/pdf/issues_pdf_helper.rb b/lib/redmine/export/pdf/issues_pdf_helper.rb
index a7a4956..a3293a3 100644
--- a/lib/redmine/export/pdf/issues_pdf_helper.rb
+++ b/lib/redmine/export/pdf/issues_pdf_helper.rb
@@ -360,6 +360,8 @@ module Redmine
   format_date(value)
   elsif value.is_a?(Time)
     format_time(value)
+  elsif value.respond_to?(:collect)
+    value.collect(&:to_s).compact.join(', ')
   else
     value
   end
end
```

How about joining value in your plugin?

#5 - 2019-03-28 23:48 - Anonymous

- Status changed from New to Resolved

Closing this as I am no longer interested in the issue.

#6 - 2019-03-29 04:48 - Marius BĂLTEANU

- Status changed from Resolved to Closed

Codruț Gușoi wrote:

Closing this as I am no longer interested in the issue.

Thanks.

Files

0001-Fixed-exporting-collections-associated-with-issues.patch	1.58 KB	2016-01-25	Anonymous
pdf_export_after.pdf	95 KB	2016-01-25	Anonymous
pdf_export_before.pdf	95.1 KB	2016-01-25	Anonymous
csv_export_after.csv	111 Bytes	2016-01-25	Anonymous
csv_export_before.csv	186 Bytes	2016-01-25	Anonymous