ChiliProject - Bug # 935: Serialization problem in Setting model

Status:	Closed	**Priority:**	Normal
Author:	Robert Mitwicki	**Category:**	
Created:	2012-03-16	**Assignee:**	Holger Just
Updated:	2012-07-11	**Due date:**	
Remote issue URL:			
Affected version:			
Description:	If You use plugin where it store serialized data sometimes (I do not know why sometimes) chiliproject can crash. This is because when CP use find_or_default method from Setting model and try to create new object. But value is setup by Active Record but he do not know that this attribute should be serialized and he convert that to string. Which follow to crash when want to get this value. It happens just when find_or_default method is used. I fixed that just replace value methods by serialize: https://github.com/mitfik/chiliproject/commit/b824548700eac441bad0e9d6ad8468465dac9b5f Pull request is waiting for Your acceptance. Additional: There was similar problem with readmine: http://www.redmine.org/projects/redmine/repository/revisions/8909		

Associated revisions

2008-03-26 11:22 pm - Jean-Philippe Lang

Fixed: when bulk editing, setting "Assigned to" to "nobody" causes an sql error with Postgresql (#935).

git-svn-id: http://redmine.rubyforge.org/svn/trunk@1294 e93f8b46-1217-0410-a6f0-8f06a7374b81

2012-07-11 12:56 pm - Romano Licker

[#935] Setting improperly set up for default values

having a fixed call order resolves a problem where
'value=' was called before 'name=' resulting in no
serialization

History

2012-03-20 09:17 pm - Felix Schäfer

Hello Robert,

Can you provide a test case or a least an example of the failure?

2012-03-20 10:51 pm - Robert Mitwicki

Felix SchÃ¤fer wrote:

> Hello Robert,
>
> Can you provide a test case or a least an example of the failure?

Hi,

Here is example with plugin chiliproject_backlogs:
https://github.com/finnlabs/chiliproject_backlogs/issues/9

But the best example is if You will add some new gem, which has some default setting options (for example chiliproject_backlogs)
then in console just type:
  Setting.plugin_backlogs
and instead of normal hash You will get string, and this is the problem because if there is no already added settings to database chiliproject use find_or_default method where is used new method from ActiveRecord which do not know that value should be serialized.

From other point I do not see any reason why in setting do not use serialize method from ActiveRecord instead of those home made methods for that. ;)

best regards

2012-03-21 07:33 pm - Felix Schäfer

- Status changed from Ready for review to Open

The proposed fix won't work though as defaults are read from a yml file as well as from the DB.

2012-03-21 09:08 pm - Robert Mitwicki

Felix SchÃ¤fer wrote:
> The proposed fix won't work though as defaults are read from a yml file as well as from the DB.

Are sure about that, maybe a little example what happen in find_or_default method which is called to get setting if does not exist it get default from yml:

```
<pre>
>> Setting.new(:name => "foo", :value => {:bar => "foo"})
=> #<Setting id: nil, name: "foo", value: "barfoo", updated_on: nil>
</pre>
```

so value is String not hash (because of missing serialize for AR model)

After my fix (remove value and value= methods):

```
<pre>
>> Setting.new(:name => "foo", :value => {:bar => "foo"})
=> #<Setting id: nil, name: "foo", value: {:bar=>"foo"}, updated_on: nil>

</pre>
```

2012-03-22 09:51 am - Robert Mitwicki

I tried to figure out why it happens like that and I found a cause.
The problem is with value=(v) method.

Simple example:

This will work:

```
<pre>
setting = Setting.new
setting.name = "plugin_foo"
setting.value = {:bar => 2}
</pre>
```

but this not:

```
<pre>
setting = Setting.new
setting.value = {:bar => 2}
setting.name = "plugin_foo"
</pre>
```

because to set up value we need first have name, without that we can't get @@available_settings[name], it is nil.
Because of that sometimes (depend of that what AR will first create name or value) he write string instead of hash to database because in value=(v) is:

```
<pre>
v = v.to_yaml if v && @@available_settings[name]
write_attribute(:value, v.to_s)
</pre>
```

so in that case v will be not converted to yaml and it will convert hash to string and store in database.

This is why remove value=(v) and value method fix the problem. I think this is the best solution to replace them by serialize from AR, it make code cleaner and do the same things.

Links:
pull request: https://github.com/chiliproject/chiliproject/pull/167

2012-03-22 10:55 am - Felix Schäfer

Robert Mitwicki wrote:
> Felix SchÃ¤fer wrote:
> > The proposed fix won't work though as defaults are read from a yml file as well as from the DB.
>
> Are sure about that, maybe a little example what happen in find_or_default method which is called to get setting if does not exist it get default from yml:

You are right, the @value@ and @value=@ don't read values from the settings.yml file, they do read formats from it though. Sorry about that.

Back to the problem, I don't think it was ever intended for those 2 methods to be used directly, access to the settings should all go through the @Setting@ class with just @Setting.some_setting@ and @Setting.some_setting=@, so I'm unsure about the best way to handle this.

IIRC Holger was the last one to work on the settings and had some ideas for future updates to it, so I'd like him to have a look too.

2012-03-25 12:09 pm - Felix Schäfer

- Assignee changed from Robert Mitwicki to Holger Just

- Status changed from Open to Ready for review

Regarding an example of why serialize doesn't work the same as the current behavior: for keys defined as symbols in the settings.yml, the value is coerced to a symbol on read:

```
<pre>>> Setting.user_format = "bar"
=> "bar"
>> Setting.user_format
=> :bar</pre>
```

I don't say this is the best behavior and frankly I'd probably be surprised myself to store a string and get a symbol out (especially because this is only the case for settings defined as symbols in the settings.yml!!), but it's a difference nonetheless. As @user_format@ is the only one affected and shouldn't ever receive anything else than a symbol, this is probably a non-problemâ€¦

I tested the other formats too and couldn't find a difference in behavior between the current and the version with @serialized@, so although this would introduce a (minor) change in behavior in a minor release, I'd be willing to merge it, I'd still like one of the others to take a look at it though. I've created a new clean pull request at https://github.com/chiliproject/chiliproject/pull/173.

2012-07-11 12:55 pm - Gregor Schmidt
Just wanted to add my 2 Cents here.

I was running into this while working at Finnlabs, we fixed it in the finnlabs/chiliproject branches and now I'm running into it again. It seems to be related to the fact, that Hash ordering in 1.8.7 is not specified.

When initializing a new Setting instance via @.new(attributes)@ ActiveRecord uses the defined getters and setters to set the attributes. When setting name and value in one call, it is not defined, if the name or the value is set first. If the value is set before the name, then the checks within @value=@ if the attribute should be serialized or not, do not work.

I do not know under which circumstances the one or the other hash order occurs. All I know is, that sometimes it just goes wrong.

I will add a pull request which simply changes the way, the Setting model is initialized, but I agree with the OP, that we should eventually use ActiveRecord's built-ins.

2012-07-11 01:01 pm - Gregor Schmidt
You may find my Pull Request at https://github.com/chiliproject/chiliproject/pull/222

Gregor

2012-07-11 01:20 pm - Holger Just
- Status changed from Ready for review to Closed

- Target version set to 3.3.0

Thanks, Gregor. I like the simplicity of your patch.

Compared to the previous pull request, it also retains the data structure of the saved data. I was a bit unsure about the final semantics of forcing the serializable to the @value@ column already containing data.

I just merged Gregor's patch in commit:1e96ac9.