

ChiliProject - Bug # 1206: ActiveRecord double-escaping `settings` value

Status:	Open	Priority:	Normal
Author:	Will Maier	Category:	
Created:	2013-01-16	Assignee:	Felix Schäfer
Updated:	2013-01-16	Due date:	

Remote issue URL:

Affected version:

Description:

I'm testing the recent 3.5.0 chiliproject release in a Vagrant VM using the very awesome chiliproject Chef cookbook from <https://github.com/chiliproject/cookbook>. My test installation includes a single chili instance with data like this:

```
<pre>
{
  "id": "simple",
  "base_uri": "https://chili.example.com/",
  "session": {
    "secret": "MYSECRET"
  },
  "database": {
    "user": "simple_chili",
    "database": "simple_chili",
    "password": "MYPASSWORD"
  },
  "apache": {
    "ssl_certificate_file": "ssl/chili.example.com.pem",
    "ssl_key_file": "ssl/chili.example.com.key"
  },
  "ignored_bundler_groups": ["rmagick"]
}
</pre>
```

I'm using a Chef recipe along the following lines to get everything installed:

```
<pre>
include_recipe "build-essential::default"

package "libcurl4-openssl-dev"
package "ruby1.9.3"

data_bag("chiliproject").each do |name|
  project = data_bag_item("chiliproject", name)
  apache = project['apache'] || {}

  files = [
    apache["ssl_certificate_file"],
    apache["ssl_key_file"],
  ]

  files.each do |fname|
    next if fname.nil?

    path = File::join(node['apache']['dir'], fname)
```

```

cookbook_file path do
  source fname
  owner node['apache']['user']
  mode "0400"
end

end

end

include_recipe "chiliproject::default"
include_recipe "chiliproject::apache2"
include_recipe "postgresql::server_debian"
include_recipe "postgresql::server"
include_recipe "chili_simple::database"
include_recipe "chili_simple::chili"
</pre>

```

After the Vagrant `chef-solo` provider converges, though, attempts to connect to the running apache2 service produce the following error in the chili logs:

```

<pre>
Processing WelcomeController#index (for 10.0.2.2 at 2013-01-14 22:13:27) [GET]
  Parameters: {"controller"=>"welcome", "action"=>"index"}

ActiveRecord::StatementInvalid (PG::Error: ERROR: relation "settings" does not exist
LINE 4:         WHERE a.attrelid = "settings"::regclass
                        ^
):
  SELECT a.attname, format_type(a.attypid, a.atttypmod), d.adsrc, a.attnotnull
 FROM pg_attribute a LEFT JOIN pg_attrdef d
   ON a.attrelid = d.adrelid AND a.attnum = d.adnum
 WHERE a.attrelid = "settings"::regclass
   AND a.attnum > 0 AND NOT a.attisdropped
 ORDER BY a.attnum
):

```

```

vendor/bundle/ruby/1.9.1/gems/activerecord-2.3.15/lib/active_record/connection_adapters/abstract_adapter.rb:227:in `rescue in log'

```

```

vendor/bundle/ruby/1.9.1/gems/activerecord-2.3.15/lib/active_record/connection_adapters/abstract_adapter.rb:204:in `log'

```

```

vendor/bundle/ruby/1.9.1/gems/activerecord-2.3.15/lib/active_record/connection_adapters/postgresql_adapter.rb:507:in `query'

```

```

vendor/bundle/ruby/1.9.1/gems/activerecord-2.3.15/lib/active_record/connection_adapters/postgresql_adapter.rb:1063:in `column_definitions'

```

```

vendor/bundle/ruby/1.9.1/gems/activerecord-2.3.15/lib/active_record/connection_adapters/postgresql_adapter.rb:675:in `columns'

```

```

  vendor/bundle/ruby/1.9.1/gems/activerecord-2.3.15/lib/active_record/base.rb:1305:in `columns'

```

```

  vendor/bundle/ruby/1.9.1/gems/activerecord-2.3.15/lib/active_record/calculations.rb:300:in `column_for'

```

```

  vendor/bundle/ruby/1.9.1/gems/activerecord-2.3.15/lib/active_record/calculations.rb:129:in `calculate'

```

```

  vendor/bundle/ruby/1.9.1/gems/activerecord-2.3.15/lib/active_record/calculations.rb:75:in `maximum'

```

```
app/models/setting.rb:195:in `cache_key'
app/models/setting.rb:101:in `[]'
app/models/setting.rb:137:in `rest_api_enabled?'
app/controllers/application_controller.rb:86:in `find_current_user'
app/controllers/application_controller.rb:69:in `user_setup'
</pre>
```

I'm not super familiar with Rails, but it seems to me that ActiveRecord is double-escaping what appears to be the inferred relation name for the Model controller. Is there a simple config issue I've overlooked? Thanks!

History

2013-01-16 03:50 pm - Will Maier

Here's the Vagrantfile I'm using:

```
<pre>
#!/usr/bin/env ruby
# ^^ for syntax highlighting

require 'berkshelf/vagrant'

cookbook = Chef::Cookbook::Metadata.new
begin
  cookbook.from_json(IO.read("metadata.json"))
rescue Errno::ENOENT
  cookbook.from_file("metadata.rb")
end
name = cookbook.name
hostname = name.gsub('_', '-')

box = 'opscode-ubuntu-12.04'
box_url = ENV['VAGRANT_BOX_URL'] || "https://opscode-vm.s3.amazonaws.com/vagrant/boxes/#{box}.box"

Vagrant::Config.run do |config|

  config.vm.host_name = hostname
  config.vm.box = box
  config.vm.box_url = box_url
  config.vm.network :hostonly, '33.33.33.10'
  config.vm.forward_port 443, 10808

  config.vm.provision :chef_solo do |chef|
    chef.data_bags_path = File.join(Dir.pwd, "data")
    chef.run_list = [ "recipe[#{name}::default]" ]
  end
end
end
</pre>
```

The VM runs ubuntu 12.04 x86_64.