

ChiliProject - Feature # 350: Setting model should use Rails.cache instead of class variable

Status:	Closed	Priority:	Normal
Author:	Jan Schulz-Hofen	Category:	Refactoring
Created:	2011-04-25	Assignee:	Eric Davis
Updated:	2011-06-17	Due date:	
Remote issue URL:	http://www.redmine.org/issues/8222		
Affected version:			
Description:	<p>The Setting model uses two local class variables called <code>@cached_settings</code> and <code>@cached_cleared_on</code> to prevent reloading the settings from database and/or settings.yml at every request.</p> <p>Rails has been supporting different cache implementations for some time now, why not let the Setting model use them as well? Besides the fact that this will help with tenant switching which we've implemented for Planio, it should also improve Redmine's performance when running it with more than processes and a custom cache mechanism, like memcached.</p> <p>A patch is attached.</p>		

Associated revisions

2011-05-25 05:11 pm - Felix Schäfer

Use Rails.cache to cache application settings. #350

2011-05-27 11:59 pm - Eric Davis

[#350] Refactor: extract method

2011-05-28 12:52 am - Eric Davis

[#350] Force clearing the cache before each test

2011-06-10 08:44 pm - Eric Davis

[#350] Remove freeze from Settings so the values can be modified later

History

2011-04-25 05:47 pm - Jan Schulz-Hofen

- Target version deleted (1.3.0)

2011-04-26 01:20 pm - Felix Schäfer

- Target version set to 2.0.0

- Assignee set to Felix Schäfer

I'll have a look at it, but I support the general idea (I do use memcache too ;-)). I don't see a problem including it in version#8 if the tests pass, I don't know if I'll get in there until the release though, so version#5 at the latest.

2011-04-26 05:33 pm - Jan Schulz-Hofen

Thanks for the update, Felix!

2011-04-26 10:04 pm - Eric Davis

I also like the general idea but I think it's too late for 1.3.0. Maybe 1.4.0 or 2.0.0 (depending on release dates).

2011-04-27 08:18 am - Gregor Schmidt

I also like the idea.

Concerning the patch, I dislike the usage of the 'redmine-settings'. I am not familiar with best practices concerning cache keys, but I guess we should find something better than prefixing it with 'redmine-'

2011-04-27 09:42 am - Jan Schulz-Hofen

like @chiliproject-@ ?

#SCNR# ;-)

2011-04-27 07:09 pm - Felix Schäfer

Gregor Schmidt wrote:

> Concerning the patch, I dislike the usage of the 'redmine-settings'. I am not familiar with best practices concerning cache keys, but I guess we should find something better than prefixing it with 'redmine-'

I just had a quick look, and there doesn't seem to be a really standard cache key naming mechanism save the inbuilt rails one. Rails uses the @cache_key@ method on various objects to generate cache keys, for example @ActiveRecord@ uses @"#{self.class.name.tableize}/#{id}"@, other schemes are "hostname/action" and so on. Anyway, I think we should agree on some prefix to use for custom caches, and I'd say we shouldn't risk collisions with rails internals if we prefix it with a "/", so something like @/internal/settings/#{key_name}@.

2011-05-25 01:45 pm - Felix Schäfer

Two things we've (Holger and I) noticed here: as per the "@ActiveSupport::Cache::Store@"

API":<http://rails.rubyonrails.org/classes/ActiveSupport/Cache/Store.html>, it "is meant for caching strings", and what you get from @Setting.some_value@ with the @Rails.cache@ implementation are immutable objects.

We've worked around the first problem by marshalling/unmarshalling objects as needed, the second point must be enforced by freezing the objects returned by any @Setting.some_value@ method.

The naming convention used for the cache keys is: @chiliproject/#{self.class.name.tableize}/#{key_name}@ and @chiliproject/#{self.class.name.tableize}-#{attribute_name}@. Scoping everything under @chiliproject/@ should avoid collisions with rails internals, the names are all lower-case because we couldn't find any information about the case-sensitivity of the keys in the cache API.

As soon as the tests have finished passing on my local machine, I'll create a pull request so that Eric can have a look too.

2011-05-25 03:15 pm - Felix Schäfer

- Assignee deleted (Felix Schäfer)

The pull request is at <https://github.com/chiliproject/chiliproject/pull/68>

One note: for some reason the @issues_controller_test.rb@ has some errors when run from @test:functionals@ but not when run standalone, not sure if this comes from these changes or something else.

2011-05-27 09:47 pm - Eric Davis

- Assignee set to Eric Davis

The pull request looks good. I'm going to pull it in and do a touch of refactoring on it. @Setting@ could still use some cleanup but this is heading in a great direction.

2011-05-27 10:42 pm - Jan Schulz-Hofen

- Status changed from Ready for review to Open

Been discussing this with Felix the other night: Core may not need to modify settings, but plugins do, so it would be wise to @.dup@ the setting values after fetching them from the cache. I know a number of plugins which would break using the current solution.

2011-05-27 11:24 pm - Eric Davis

- Status changed from Open to Ready for review

I've merged Felix's pull request with a few changes (one that brute-force fixes the tests).

I talked with Jan on IRC and what he is trying to do is set a key on a serialized Setting. I don't remember it working before but if we can make it work then that could make a lot of things easier for me too. Feel free to make it a separate issue if you don't think it's caused by this code.

```
<pre><code class="ruby">
Setting.plugin_my_plugin[:foo] = 'bar'
</code></pre>
```

Failing test to illustrate

```
<pre><code class="diff">
diff --cc test/unit/setting_test.rb
index b63e7de,b63e7de..d19b187
--- a/test/unit/setting_test.rb
+++ b/test/unit/setting_test.rb
@@@ -42,4 -42,4 +42,14 @@@ class SettingTest < ActiveSupport::Test
  assert_equal ['issue_added', 'issue_updated', 'news_added'], Setting.notified_events
  assert_equal ['issue_added', 'issue_updated', 'news_added'], Setting.find_by_name('notified_events').value
end
++
++ test "setting a deep hash" do
++   Setting.notified_events = {:one => 1, :two => 2}
++   assert_equal 1, Setting.notified_events[:one]
++   assert_equal 2, Setting.notified_events[:two]
++
++   Setting.notified_events[:three] = 3
++   assert_equal 3, Setting.notified_events[:three]
++ end
++
</code></pre>
```

2011-05-28 02:29 pm - Jan Schulz-Hofen

Did you leave out the @.dup@ on purpose?

2011-05-29 06:29 pm - Eric Davis

Jan Schulz-Hofen wrote:

> Did you leave out the @.dup@ on purpose?

Yes, I just applied the original code. Holger also posted some reasons for the freeze to IRC so I'll let him speak up (I was away).

2011-05-30 08:26 am - Holger Just

I talked to Jan again yesterday and we think that the original (unfrozen) behavior should be retained for now, as changing that would break any plugins which set parts of complex settings today. As a conclusion, we should skip on the freeze completely for now until we have a real solution.

2011-05-30 08:28 am - Jan Schulz-Hofen

...and also add @.dup@ at the end as some cache implementations (e.g. memory store) already return a frozen hash...

2011-05-30 08:38 am - Holger Just

Jan Schulz-Hofen wrote:

> ...and also add @.dup@ at the end as some cache implementations (e.g. memory store) already return a frozen hash...

This isn't necessary as we pipe ANY read (in the @self.[](name)@ method) through @Marshal.load@ which always returns unfrozen objects.

2011-05-30 08:59 am - Jan Schulz-Hofen

My bad. Agree.

2011-05-30 02:45 pm - Felix Schäfer

- Assignee changed from Eric Davis to Felix Schäfer

- Status changed from Ready for review to Open

I'll update my pull request accordingly.

2011-06-10 07:05 pm - Eric Davis

- Status changed from Open to Ready for review

Felix:

Is this all that is needed to close this out? <https://github.com/chiliproject/chiliproject/pull/74>

2011-06-17 10:03 pm - Eric Davis

- Assignee changed from Felix Schäfer to Eric Davis

- Status changed from Ready for review to Closed

Merged into release-v2.0.0. If you want to make any changes, lets start a new issue for them.

Files

setting_rb_use_rails_cache.patch	1.7 kB	2011-04-25	Jan Schulz-Hofen
----------------------------------	--------	------------	------------------