

Redmine - Defect #11141

Redmine should have 'minitest' in its Gemfile

2012-06-12 08:19 - Bohuslav Kabrda

Status: Closed	Start date:
Priority: Normal	Due date:
Assignee:	% Done: 0%
Category:	Estimated time: 0.00 hour
Target version:	Affected version:
Resolution:	
Description Hi, I think it would be a good idea to add "gem 'minitest'" into your Gemfile. Some distributions, like Fedora 17, have unbundled the gems from MRI Ruby and therefore these gems can't be loaded now, if not present in Gemfile. (Please note, that I'm well aware of #11071 and this has already been solved on Fedora 17. This is something different :)) Thanks!	
Related issues: Has duplicate Redmine - Defect #11281: Add minitest to the GemFile dependency Closed	

History

#1 - 2012-06-12 23:19 - Jean-Philippe Lang

I disagree. Redmine uses standard Rails test classes (ActiveSupport::TestCase and its subclasses) that inherits from Test::Unit::TestCase. Why the Redmine Gemfile should be aware that these classes are on top of MiniTest in ruby1.9?

#2 - 2012-06-13 07:25 - Bohuslav Kabrda

Here are few reasons:

- My main reason is that when the minitest gem is unbundled from Ruby, the failure occurs. I agree that this is a distro specific issue, so here are few more general:
- By not having minitest in Gemfile, you don't know which version of minitest will get into play - different versions/patchlevels of ruby 1.9.x contain different versions of minitest. Isn't this the reason of having Gemfile - to lock your dependencies? How can you be sure that newer version with some backward incompatibilities of isn't in any of the mentioned Rubies?
- AFAIK, when using Gemfile, it is customary to include all the dependencies that the app needs (and that aren't drawn in automatically by other dependencies). What is the reason of specialcasing minitest?
- Another thing is, that when doing "RAILS_ENV=production rake db:migrate", minitest shouldn't be needed at all, I don't really see the purpose of requiring it during db migration.

#3 - 2012-06-13 08:58 - Jean-Philippe Lang

Bohuslav Kabrda wrote:

Here are few reasons:

- My main reason is that when the minitest gem is unbundled from Ruby, the failure occurs. I agree that this is a distro specific issue

Very true, it's a distro specific issue and I don't want to include distro specific stuff in Redmine.

- By not having minitest in Gemfile, you don't know which version of minitest will get into play - different versions/patchlevels of ruby 1.9.x contain different versions of minitest. Isn't this the reason of having Gemfile - to lock your dependencies? How can you be sure that newer version with some backward incompatibilities of isn't in any of the mentioned Rubies?

Minitest **is not** a direct Redmine dependency, Redmine depends on Rails for testing and uses its API. The rails version that I'm using is locked in the Gemfile (eg. 3.2.5 today) but I don't care about the version of MiniTest. I just care about the Rails API.

- AFAIK, when using Gemfile, it is customary to include all the dependencies that the app needs (and that aren't drawn in automatically by other dependencies). What is the reason of specialcasing minitest?

Do you really think that I've added all the Rails dependencies in the Redmine Gemfile? Please have a look at Gemfile.lock.

- Another thing is, that when doing "RAILS_ENV=production rake db:migrate", minitest shouldn't be needed at all, I don't really see the purpose of requiring it during db migration.

Again, I don't require 'minitest' anywhere. So I think you're gonna have this issue with any Rails app.

#4 - 2012-06-13 09:24 - Bohuslav Kabrda

Jean-Philippe Lang wrote:

Bohuslav Kabrda wrote:

Here are few reasons:

- My main reason is that when the minitest gem is unbundled from Ruby, the failure occurs. I agree that this is a distro specific issue

Very true, it's a distro specific issue and I don't want to include distro specific stuff in Redmine.

- By not having minitest in Gemfile, you don't know which version of minitest will get into play - different versions/patchlevels of ruby 1.9.x contain different versions of minitest. Isn't this the reason of having Gemfile - to lock your dependencies? How can you be sure that newer version with some backward incompatibilities of isn't in any of the mentioned Rubies?

Minitest **is not** a direct Redmine dependency, Redmine depends on Rails for testing and uses its API. The rails version that I'm using is locked in the Gemfile (eg. 3.2.5 today) but I don't care about the version of MiniTest. I just care about the Rails API.

Yes, but still Redmine can't run without being able to require 'test/unit' (be it in whatever testing library).

- AFAIK, when using Gemfile, it is customary to include all the dependencies that the app needs (and that aren't drawn in automatically by other dependencies). What is the reason of specialcasing minitest?

Do you really think that I've added all the Rails dependencies in the Redmine Gemfile? Please have a look at Gemfile.lock.

Have you read the part "and that aren't drawn in automatically by other dependencies"? :)

- Another thing is, that when doing "RAILS_ENV=production rake db:migrate", minitest shouldn't be needed at all, I don't really see the purpose of requiring it during db migration.

Again, I don't require 'minitest' anywhere. So I think you're gonna have this issue with any Rails app.

Ah, not actually your problem (and neither mine). The problem is in prototype-rails which deals with `ActionView::TestCase.class_eval` in the library itself.

#5 - 2015-10-06 01:18 - Toshi MARUYAMA

- *Description updated*

#6 - 2015-12-11 14:05 - Toshi MARUYAMA

- *Status changed from New to Closed*

2.6-stable (Rails 3.2): [r14181](#).

Rails 4.2 requires minitest.

<https://github.com/rails/rails/blame/v4.2.5/activerecord/activerecord.gemspec#L26>