

ChiliProject - Feature # 269: Refactor lib/redmine/menu_manager.rb to increase extensibility

Status:	Closed	Priority:	Normal
Author:	Gregor Schmidt	Category:	Refactoring
Created:	2011-03-09	Assignee:	Eric Davis
Updated:	2011-03-24	Due date:	

Remote issue URL:

Affected version:

Description: Currently, the source:lib/redmine/menu_manager.rb is a rather large file containing multiple inner classes and modules of MenuManager as well as patches for rubytree. Concerning Rails' reloading mechanisms, this ties all these things into one big clump.

I was trying to extend the `@Redmine::MenuManager::MenuHelper@` from within a plugin, but was repetitively running into reloading issues.

At first the repeated loading of file:lib/redmine/menu_manager.rb patching of the rubytree gem resulted in stack level to deep errors (see #266). To cut a long story short. The only way to fix my problem, is to extract all building blocks of the MenuManager into separate files, to allow reloading them on every request during development.

I am therefore proposing to refactor the menu_manager.rb into separate files and, while we're at it, remove the monkey patching of rubytree.

Associated revisions

2011-03-09 04:02 pm - Gregor Schmidt

[#269] Extracting classes and modules from menu manager into separate files

2011-03-09 04:26 pm - Gregor Schmidt

[#269] Instead of patching rubytree, creating a custom sub class and using that everywhere

History

2011-03-09 03:10 pm - Gregor Schmidt

Downside of this approach would be an increased cost when applying redmine patches to the chili sources.

Upside would be, that this refactoring increases the extensibility of chili through plugins.

2011-03-09 03:42 pm - Gregor Schmidt

- Assignee deleted (Gregor Schmidt)

- Status changed from Open to Ready for review

The pull request is sitting at <https://github.com/chiliproject/chiliproject/pull/19>.

2011-03-18 09:24 am - Gregor Schmidt

I would be happy to hear any feedback on this topic.

2011-03-18 11:53 pm - Eric Davis

Your pull request removed the bundled gem without adding it's requirement back in anywhere (e.g. no `@config.gem@`). I think removing the bundled gem should be separate from splitting the classes.

2011-03-19 09:19 am - Gregor Schmidt

Eric Davis wrote:

> Your pull request removed the bundled gem without adding its requirement back in anywhere (e.g. no `@config.gem@`).

The `@config.gem@` line is in `@config/environment.rb@` ever since. Therefore there was no need to add it.

> I think removing the bundled gem should be separate from splitting the classes.

I totally agree with that. I was a bit over-the-top with that one. Sorry for that. I've edited the request accordingly.

Thanks for taking the time and giving the feedback.

Gregor

2011-03-19 07:39 pm - Eric Davis

- *Target version set to 2.0.0*

- *Assignee set to Eric Davis*

- *Category set to Refactoring*

- *Status changed from Ready for review to Closed*

Gregor Schmidt wrote:

> The `@config.gem@` line is in `@config/environment.rb@` ever since. Therefore there was no need to add it.

My mistake, sorry. I forgot that `@vendor/gems@` aren't autoloaded like plugins.

I've merged your changes into unstable. I like what you did and would like to see more of `@lib/@` split like that.

2011-03-19 09:24 pm - Gregor Schmidt

Thanks. I'm happy to help.

Gregor

2011-03-24 09:52 am - Gregor Schmidt

For future reference, if anybody happens to come here using search or something:

The gem line in `config/environment.rb` which defines the dependency to `rubytree` is not specific enough. It should be something like

```
<pre>
  config.gem 'rubytree', :lib => 'tree', :version => '~> 0.5.2'
</pre>
```