

## Tree based menus

2022/05/09 14:19 - Admin Redmine

ステータス:	Closed	開始日:	2009/11/19
優先度:	通常	期日:	
担当者:		進捗率:	100%
カテゴリ:	UI_10	予定工数:	0.00時間
対象バージョン:	0.9.0_6	作業時間:	0.00時間
Redmineorg_URL:	https://www.redmine.org/issues/4250	status_id:	5
category_id:	10	tracker_id:	3
version_id:	6	plus1:	1
issue_org_id:	4250	affected_version:	
author_id:	5	closed_on:	
assigned_to_id:	5	affected_version_id:	
comments:	14		

### 説明

Currently Redmine's menus are only liner: Overview, Activity...

These patches convert the menus into a tree structure. The first patch will let menus be grouped and organized like so:

- Overview
- Issues \*\* New Issue
- Activity
- Settings

Which renders to html as:

```
... other menu items here ...  
• Issues  
  ◦ New issue  
... other items ...
```

The second patch will allow a menu to define an "unattached" menu. This type of menu is a chunk of code that will create a menu on the fly, based on the current page's data (e.g. current project). I've used this to add a menu that lists all of the Custom Queries on a project:

```
... other menu items here ...  
• Issues  
  ◦ New issue  
  ◦ Late
```

[Needs an estimate](#)

... other items ...

These two patches (plus some javascript and custom css) are what makes the "Shane and Peter theme":  
<http://www.redmine.org/boards/1/topics/4402> work. You can see how the theme is building the menus in it's "init.rb":  
[http://github.com/edavis10/redmine\\_shane\\_and\\_peter\\_design/blob/master/init.rb#L32](http://github.com/edavis10/redmine_shane_and_peter_design/blob/master/init.rb#L32)

Both of these patches are backwards compatible. They only change how menus are stored internally and provide a some new methods for core/plugin menus. I've also included a bunch of tests for the MenuManager (which was untested). Once I get a code review or two, I can commit these. If anyone has any questions, feel free to ask.

journals

Not a real code review but looking at the patches this seems to be a huge improvement over the current implementation of the menus in Redmine.

+1

This goes a long way too in the ability to support plugins so that they may configure and add custom menu items. for example,

- listing watched wiki pages in a drop down under wiki
- showing all settings in the drop menu to avoid having to click 2x just to get to members
- create a reports tab that includes both default reporting items and new custom reports

etc...

+1

+1

Nice improvement for plugins indeed.

Small details:

- the existing doc should be updated (eg. new option :parent\_menu in Mapper#push is absent)
- we have :before, :after, :last options (and not :before\_menu, ...) so I expect a :parent option instead of :parent\_menu
- same thing for MenuItem#new, :parent\_menu and :child\_menus should be renamed to :parent and :children

\* it uses few methods of the (unpatched) rubytree lib, thus I'm not sure if it's worth the gem requirement

Can't wait.

I've been trying this patch on my system for several weeks now and it's been really great. No problems and our team is working more efficiently. Thanks Eric!!

Our whole team loves the new menus. I can't wait to see this implemented so that the community can start coding to it.

Jean-Philippe Lang wrote:

Small details:

- the existing doc should be updated (eg. new option :parent\_menu in Mapper#push is absent)

Agreed.

- we have `:before`, `:after`, `:last` options (and not `:before_menu`, ...) so I expect a `:parent` option instead of `:parent_menu`
- same thing for `MenuItem#new`, `:parent_menu` and `:child_menus` should be renamed to `:parent` and `:children`

I agree but I think there was a reason I didn't use `@:parent@`. I think the node class defined and used `@parent@` for something else. I'll look into it and see.

- it uses few methods of the (unpatched) `rubytree` lib, thus I'm not sure if it's worth the gem requirement

What do you propose?

Bundle the gem with Redmine in vendor?

Extract only the classes we use and bundle them with Redmine in vendor?

Patch the gem and require a custom patched version? (I could host the gem on my Rubyforge/gemcutter account)

---

Here's a new patch (3 in the series) with a few updates:

- Updated `Mapper#push` documentation
- Renamed `:parent_menu` to `:parent`
- Renamed the external API for `:child_menus` to `:children`. Internally it needs to stay `:child_menus` because `Tree::TreeNode` already defines a `children` method for another purpose

The only other thing remaining is to decide on the `rubytree` gem dependency. I think it would be ok to include it into vendor, it's only 2,058 lines of code (1242 in `setup.rb`, 554 in tests, 262 in lib).

Eric Davis wrote:

Here's a new patch (3 in the series) with a few updates:

- Updated `Mapper#push` documentation
- Renamed `:parent_menu` to `:parent`
- Renamed the external API for `:child_menus` to `:children`. Internally it needs to stay `:child_menus` because `Tree::TreeNode` already defines a `children` method for another purpose

Thanks.

The only other thing remaining is to decide on the `rubytree` gem dependency. I think it would be ok to include it into vendor, it's only 2,058 lines of code (1242 in `setup.rb`, 554 in tests, 262 in lib).

Agreed.

Since this is in support of the SP theme, it has my vote.

Added to trunk in r3090, r3091, and r3092. Thanks for the feedback and reviews everyone.

---

related\_issues

relates,Closed,6427,Create CSS to support child and nested menus

履歴

#1 - 2022/05/10 17:24 - Admin Redmine

- カテゴリをUI\_10にセット

- 対象バージョンを0.9.0\_6にセット