Redmine - Patch #4250

Tree based menus

2009-11-20 00:02 - Eric Davis

Status:	Closed		**Start date:**	2009-11-19
Priority:	Normal		**Due date:**	
Assignee:	Eric Davis		**% Done:**	100%
Category:	UI		**Estimated time:**	0.00 hour
Target version:	0.9.0			

Description

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:

```
<div id="main-menu">
  <ul>
    ... other menu items here ...
    <li>
      <a class="issues" href="/projects/abc/issues">Issues</a>
      <ul class="menu-children">
        <li>
          <a class="new-issue" accesskey="7" href="/projects/abc/issues/new">New issue</a>
        </li>
      </ul>
    </li>
    ... other items ...
  </ul>
</div>
```

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:

```
<div id="main-menu">
  <ul>
    ... other menu items here ...
    <li>
      <a class="issues" href="/projects/abc/issues">Issues</a>
      <ul class="menu-children">
        <li>
          <a class="new-issue" accesskey="7" href="/projects/abc/issues/new">New issue</a>
        </li>
      </ul>
      <ul class="menu-children unattached">
        <li>
          <a class="query-2" href="/projects/cremin4/issues?query_id=2">Late</a>
          <a class="query-3" href="/projects/cremin4/issues?query_id=3">Needs an estimate</a>
        </li>
      </ul>

    </li>
    ... other items ...
  </ul>
</div>
```

These two patches (plus some javascript and custom css) are what makes the [Shane and Peter theme](#) work. You can see how the theme is building the menus in it's [init.rb](#)

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.

Related issues:

Related to Redmine - Patch #6427: Create CSS to support child and nested menus	**Closed**	**2010-09-17**

Associated revisions

Revision 3090 - 2009-11-25 06:36 - Eric Davis

Converted Menus to a Tree structure to allow submenus.

- Bundle the rubytree gem
- Patched RubyTree's TreeNode to add some additional methods.
- Converted the menu rendering to walk the Tree of MenuItems to render each item
- Added a menu option for :parent_menu to make this menu a child of the parent
- Added a bunch of tests
- Made MenuItem a subclass of Tree::TreeNode in order to use it's methods directly
- Changed the exceptions in MenuItem#new to be ArgumentErrors instead of the generic RuntimeError

  #4250

Revision 3091 - 2009-11-25 06:36 - Eric Davis

Add support for unattached menus (generated dynamically)

A MenuItem can define a :child_menus option with a Proc. When the menus are rendered, the Proc will be run and the resulting MenuItems will be added to the page as child menus

`#4250`

Revision 3092 - 2009-11-25 06:36 - Eric Davis

Updated menus from JPL's feedback.

- 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

  #4250

History

#1 - 2009-11-20 00:12 - Mischa The Evil

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

#2 - 2009-11-20 00:40 - Shane Pearlman

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

#3 - 2009-11-20 12:22 - Henrik Ammer

+1

#4 - 2009-11-20 13:40 - Jean-Philippe Lang

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

#5 - 2009-11-20 16:46 - Jean-Philippe Lang

- Category set to UI

#6 - 2009-11-21 07:39 - Dan Cameron

Can't wait.

#7 - 2009-11-21 15:28 - claire s

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!!

#8 - 2009-11-22 01:30 - Peter Chester

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

#9 - 2009-11-23 19:11 - Eric Davis

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?

1. Bundle the gem with Redmine in vendor?
2. Extract only the classes we use and bundle them with Redmine in vendor?
3. Patch the gem and require a custom patched version? (I could host the gem on my Rubyforge/gemcutter account)

#10 - 2009-11-23 21:26 - Eric Davis

- File 0003-Updated-menus-from-JPL-s-feedback.patch added

- Status changed from New to 7

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).

#11 - 2009-11-23 21:53 - Jean-Philippe Lang

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.

#12 - 2009-11-24 16:11 - Ted Lilley

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

#13 - 2009-11-25 06:31 - Eric Davis

- Status changed from 7 to Closed

- Target version set to 0.9.0

- % Done changed from 0 to 100

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

#14 - 2013-07-23 21:57 - Mischa The Evil

- Related to Patch #6427: Create CSS to support child and nested menus added

Files

0001-Converted-Menus-to-a-Tree-structure-to-allow-submenu.patch	34.3 KB	2009-11-19	Eric Davis
0002-Add-support-for-unattached-menus-generated-dynamical.patch	13.1 KB	2009-11-19	Eric Davis
0003-Updated-menus-from-JPL-s-feedback.patch	15 KB	2009-11-23	Eric Davis