

Redmine - Feature #2179

Alternate layout for one site

2008-11-13 17:20 - Patrick Naubert

Status:	Closed	Start date:	2008-11-13
Priority:	Normal	Due date:	
Assignee:	Eric Davis	% Done:	0%
Category:	UI	Estimated time:	0.00 hour
Target version:			
Resolution:			
Description Just like Redmine permits Themes that modify the current layout, we would like to be able to modify the site layout. Currently, we handrolically modify the base.rhtml file. I think it would be a better idea to make the lines in app/controllers/application.rb: class ApplicationController < ActionController::Base layout 'base' To instead read from a Settings. The problem is that we are creating a redmine site for our government, and the law requires us to provide a Common Look and Feel to all government sites. (Top menu on the side instead, government banner, etc...)			

History

#1 - 2008-11-13 17:21 - Patrick Naubert

Can't believe a screwed up the feature request title...

Should be "Multiple Layouts for one site"

sigh

#2 - 2008-11-13 17:47 - Jean-Philippe Lang

- Subject changed from Umultiple Layouts for one site to Multiple Layouts for one site

Subject fixed. Anyway, I can't see why you need **multiple** layouts. Do you really need multiple layouts or just need to replace the default one without hecking base.rhtml ?

#3 - 2008-11-13 18:40 - Patrick Naubert

Yeah, I was thinking about that after I submitted. I guess just not nuking the base.rhtml is key.

The main problem is have is during development of the new layout, if I cannot easily switch back to base, I can't really progress.

Not only that, but I find that it would have to be linked to a theme too, because I had to create a theme as well (New stylesheet). If I change back to my .rhtml file without also swithcing to the new theme, I have great difficulties.

Of course, changing the application.rb file means that I have to restart mongrel. That's a pain too.

I am using SVN version of redmine.

#4 - 2008-11-13 20:20 - Eric Davis

I just tried this and I think it will be easy for you to develop and also make it easy to package custom layouts.

The engines plugin allows you to override views in plugins. Just mirror the view structure and the plugin's view will be loaded first:

```
# vendor/plugin/app/layouts/base.rhtml
<html>
  <body>
    This is my custom layout with nothing inside.
```

```
</body>
</html>
```

So you can just revert base.rhtml to the svn version and copy your custom base.rhtml into a new plugin and hack away. This technique also works for any other view file: including standard rhtml, partials, rjs, PDF views, and other plugin views.

#5 - 2008-11-16 12:35 - Jean-Philippe Lang

- Subject changed from *Multiple Layouts for one site* to *Alternate layout for one site*
- Status changed from *New* to *Closed*

Eric is right. You can achieve this by overrinding base.rhtml in a plugin.

#6 - 2008-11-18 19:18 - Patrick Naubert

- Status changed from *Closed* to *Reopened*

My Appologies for being thick, but I don't yet understand the directory structure I must follow.

Here are the directories I tried with no success:

redmine/vendor/plugins/myapp/app/layouts/base.rhtml

redmine/vendor/myapp/app/layouts/base.rhtml

redmine/vendor/plugins/engines/myapp/app/layouts/base.rhtml

Thanks muchly.

#7 - 2008-11-18 22:43 - Eric Davis

- File *redmine_plugin_layout_example.tar.gz* added
- Status changed from *Reopened* to *Closed*

Patrick Naubert wrote:

My Appologies for being thick, but I don't yet understand the directory structure I must follow.

Here are the directories I tried with no success:

redmine/vendor/plugins/myapp/app/layouts/base.rhtml

Almost got it: redmine/vendor/plugins/myapp/ **views** /layouts/base.rhtml

I've attached a plugin (redmine_foo) that should this. Just extract it into vendor/plugins. You should see a directory called REDMINE_ROOT/vendor/plugins/redmine_foo that has the init.rb. Let me know if you need anything else.

#8 - 2009-02-26 19:07 - Patrick Naubert

- Status changed from *Closed* to *Reopened*
- Assignee set to *Eric Davis*

That worked well. I am now at a later stage where I have to add 2 methods to the Redmine app/controllers/application.rb and I also have to change the set_localization method of that same file.

I tried to just create a myplugin/app/controllers/application.rb file with the following content, but it doesn't seem to use it:

```
class ApplicationController < ActionController::Base

  logger.info "In New application controller"

  alias :old_set_localization :set_localization

  def french
    cookies[:language] =
      { :value => 'fr',
        :expires => Time.now + 1.year,
        :path => '/',
        :domain => 'testredmine.ircan.gc.ca' }
    redirect_to request.referer end
end
```

```

def english
  cookies[:language] =
    { :value => 'en',
      :expires => Time.now + 1.year,
      :path => '/',
      :domain => 'testredmine.ircan.gc.ca' }
  redirect_to request.referer
end

def set_localization
  logger.info "In new set_localization"
  if url_for(params) =~ /\french$/
    french()
  end
  if url_for(params) =~ /\fenglish$/
    english()
  end
  if cookies[:language]
    User.current.language = cookies[:language]
    current_language = cookies[:language]
  else
    User.current.language = nil unless User.current.logged?
  end
  lang = begin
    if !User.current.language.blank? && GLoc.valid_language?(User.current.language)
      User.current.language
    elsif request.env['HTTP_ACCEPT_LANGUAGE']
      accept_lang = parse_qvalues(request.env['HTTP_ACCEPT_LANGUAGE']).first.downcase
      if !accept_lang.blank? && (GLoc.valid_language?(accept_lang) || GLoc.valid_language?(accept_lang = a
ccept_lang.split('-').first))
        User.current.language = accept_lang
      end
    end
  rescue
    nil
  end || Setting.default_language
  set_language_if_valid(lang)
end

```

I have a view defined in the myplugin/ plugin and that works well, so I know that my plugin hierarchy is correct, as you have given me.

#9 - 2009-02-27 22:32 - Eric Davis

- Status changed from Reopened to Closed

You can't override controllers or models that way. The best way is to create a module and include it into the base class. Here's an example for a [model](#) and a [helper](#). I haven't had to add anything to a controller yet but it's the same basic process.

If you need some more help, we should take the conversation to the forums so the issue tracker isn't cluttered.

Files

redmine_plugin_layout_example.tar.gz	988 Bytes	2008-11-18	Eric Davis
--------------------------------------	-----------	------------	------------