redmineorg-copy202205 - Vote #71928

Doesn't patch "User" model in a plugin

2022/05/09 16:12 - Admin Redmine

ステータス:	New	**開始日:**	2022/05/09
優先度:	通常	**期日:**	
担当者:		**進捗率:**	0%
カテゴリ:	Plugin API_20	**予定工数:**	0.00時間
対象バージョン:		**作業時間:**	0.00時間
Redmineorg_URL:	https://www.redmine.org/issues/11035	**status_id:**	1
category_id:	20	**tracker_id:**	1
version_id:	0	**plus1:**	0
issue_org_id:	11035	**affected_version:**	
author_id:	46901	**closed_on:**	
assigned_to_id:	0	**affected_version_id:**	43
comments:	24		

説明

Sorry for my English :)

I have a plugin that works in "Redmine 1.3.0.stable.24228 (MySQL)"
This plugin extend a model "User". And it worked fine.

Now we have installed Redmine 2.0.0
I try rewrite my plugin, but extending model "User" call error "Expected /usr/share/srv-redmine/redmine-2.0/app/models/user.rb to define User"

This init.rb code:

```
Redmine::Plugin.register :advanced_roadmap do
    name 'Advanced Roadmap plugin'
    author 'Author name'
    description 'This is a plugin for Redmine'
    version '0.0.1'
    url 'http://example.com/path/to/plugin'
    author_url 'http://example.com/about'
end

#require 'user'

ActionDispatch::Callbacks.to_prepare do
        User.send(:include, AdvancedRoadmap::UserPatch)
end
```

If I add

```
require 'user'
```

I have error:

```
ActionView::Template::Error (undefined method `association_class' for nil:NilClass):
        40:                 <%= hidden_field_tag(controller.default_search_scope, 1, :id => nil) if controller.default_search_sc
        41:
        42:                     <%= link_to l(:label_search), {:controller => 'search', :action => 'index', :id => @project},
earch) %>:
        43:
        44:                 <%= text_field_tag 'q', @question, :size => 20, :class => 'small', :accesskey => accesskey(:qu
        45:                 <% end %>
        46:                 <%= render_project_jump_box %>
```

```
    app/helpers/application_helper.rb:240:in   `render_project_jump_box'
    app/views/layouts/base.html.erb:43:in   `_app_views_layouts_base_html_erb___374889370_87506190'
    app/controllers/projects_controller.rb:168:in   `show'
```

This's content of file "lib/advanced_roadmap/user_patch.rb"

```
require_dependency   "user"

module   AdvancedRoadmap
    module   UserPatch
        def   self.included(base)
                base.extend(ClassMethods)
                base.send(:include,   InstanceMethods)

            base.class_eval   do
            end

        end

        module   ClassMethods
        end

        module   InstanceMethods
        end

    end
end
```

It's my settings

```
root@redmine:/usr/share/srv-redmine/redmine-2.0/log#   RAILS_ENV=development   rake   about
(in   /usr/share/srv-redmine/redmine-2.0)
About   your   application's   environment
Ruby   version                              1.9.3   (i686-linux)
RubyGems   version                        1.8.24
Rack   version                                 1.4
Rails   version                               3.2.3
Active   Record   version               3.2.3
Action   Pack   version                    3.2.3
Active   Resource   version          3.2.3
Action   Mailer   version               3.2.3
Active   Support   version              3.2.3
Middleware                                       ActionDispatch::Static,   Rack::Lock,   #,   Rack::Runtime,   Rack::MethodOverride,   ActionDis
ails::Rack::Logger,   ActionDispatch::ShowExceptions,   ActionDispatch::DebugExceptions,   ActionDispatch::RemoteIp,   ActionDispatch::R
eloader,   ActionDispatch::Callbacks,   ActiveRecord::ConnectionAdapters::ConnectionManagement,   ActiveRecord::QueryCache,   Action
Dispatch::Cookies,   ActionDispatch::Session::CookieStore,   ActionDispatch::Flash,   ActionDispatch::ParamsParser,   ActionDispatch::Hea
d,   Rack::ConditionalGet,   Rack::ETag,   ActionDispatch::BestStandardsSupport,   OpenIdAuthentication
Application   root                         /usr/share/srv-redmine/redmine-2.0
Environment                                        development
Database   adapter                           mysql2
Database   schema   version          20120422150750
```

Also I use: RVM (version 3.*), OS - Ubuntu 10.10

Any help is appreciated.

journals

```
#require 'user'

ActionDispatch::Callbacks.to_prepare do
User.send(:include, AdvancedRoadmap::UserPatch)
end
```

Hm, did you try this instead:

```
require 'dispatcher'

Dispatcher.to_prepare do
    require_dependency 'user'
    User.send(:include, AdvancedRoadmap::UserPatch)
end
```

That seems pretty idiomatic to me.

If i do this

```
require 'dispatcher'

Dispatcher.to_prepare do
    require_dependency 'user'
    User.send(:include, AdvancedRoadmap::UserPatch)
end
```

I cause error

```
cannot load such file -- dispatcher
```

First code seemed OK, related to "this rails issue": https://github.com/rails/rails/issues/3847 if it is an issue; try to clear your caches as suggested?

Vladimir Pitin wrote:

> If i do this
>
> [...]
>
> I cause error
>
> [...]

In fact, 'dispatcher' does not exist anymore in rails 3.
Redmine documentation for plugin development needs to be updated.
Try this instead:

```
Rails.configuration.to_prepare do
    require_dependency 'user'
    User.send(:include, AdvancedRoadmap::UserPatch)
end
```

Fabrice ROBIN wrote:

> Redmine documentation for plugin development needs to be updated.

I don't think that Redmine documentation actually refers to the use of @dispatcher@?

Exact, I was convinced that was the case.

But it might be a good idea to add something in redmine wiki.
The to_prepare method is a must have when a plugin needs to patch redmine core,
mainly in development mode.

Here is the documentation about the to_prepare method

```
Add a preparation callback. Preparation callbacks are run before every
request in development mode, and before the first request in production
mode
```

This is described in RoR guides, mainly in http://guides.rubyonrails.org/configuring.html#initializers.

Vladimir is right when he replaces the @dispatcher@ requirement with @ActionDispatch::Callbacks.to_prepare@, see http://guides.rubyonrails.org/configuring.html#configuring-action-dispatch , this is the most accurate match.

If I try it

```
Rails.configuration.to_prepare do
    require_dependency 'user'
    User.send(:include, AdvancedRoadmap::UserPatch)
end
```

I have same error

```
Expected /usr/share/srv-redmine/redmine-2.0/app/models/user.rb to define User
```

> First code seemed OK, related to this rails issue if it is an issue; try to clear your caches as suggested?

No.How can I clear my cashes?

My folder /tmp/cache is empty. If it's meant?

Upgrade to Redmine 2.0.1 had no affect
Commands:

```
rake tmp:cache:clear
rake tmp:sessions:clear
```

had no affect, too

This works:
lib/redmine_test/patches/user_patch.rb

```
require_dependency 'principal'
require_dependency 'user'
module RedmineTest
    module Patches
        module UserPatch
            def self.included(base)
                base.send(:extend, ClassMethods)
                base.send(:include, InstanceMethods)
                base.class_eval do
                    unloadable
                end
            end

            module ClassMethods
                def echoTest
                    STDOUT.print "echo Test\n"
                    STDOUT.flush
                    nil
                end

            end
            module InstanceMethods
            end
        end
    end
end

ActionDispatch::Callbacks.to_prepare do
    User.send(:include, RedmineTest::Patches::UserPatch)
end
```

After which the User.echoTest method exists - you need to add @require_dependency 'principal'@ before @require_dependency 'user'@

init.rb

```
require 'redmine'
require 'advanced_roadmap/user_patch'
Rails.logger.info 'Starting Advanced Roadmap plugin for Redmine'

Redmine::Plugin.register :advanced_roadmap do
    name 'Advanced Roadmap plugin'
    author 'Author name'
    description 'This is a plugin for Redmine'
    version '0.0.1'
    url 'http://example.com/path/to/plugin'
    author_url 'http://example.com/about'
end
```

user_patch.rb

```
#require_dependency "user"

require_dependency 'principal'
require_dependency 'user'
module AdvancedRoadmap
        module UserPatch
            def self.included(base)
                base.send(:extend, ClassMethods)
                base.send(:include, InstanceMethods)
                base.class_eval do
                    unloadable
                end
            end

            module ClassMethods
                def echoTest
                    STDOUT.print "echo Test\n"
                    STDOUT.flush
                    nil
                end

            end

            module InstanceMethods
            end
        end
end

ActionDispatch::Callbacks.to_prepare do
    User.send(:include, AdvancedRoadmap::UserPatch)
end
```

It's not work. error's is same

```
ActionView::Template::Error (undefined method `association_class' for nil:NilClass):
        40:                    <%=  hidden_field_tag(controller.default_search_scope,  1,  :id  =>  nil)  if  controller.default_search_sc
        41:
        42:                        <%=  link_to  l(:label_search),  {:controller  =>  'search',  :action  =>  'index',  :id  =>  @project},
earch)  %>:
        43:
        44:                    <%=  text_field_tag  'q',  @question,  :size  =>  20,  :class  =>  'small',  :accesskey  =>  accesskey(:qu
        45:                    <%  end  %>
        46:                    <%=  render_project_jump_box  %>
    app/helpers/application_helper.rb:240:in  `render_project_jump_box'
    app/views/layouts/base.html.erb:43:in  `_app_views_layouts_base_html_erb__947349986_81730330'
```

Hi, looks like an un-resolved dependency on the project model (via member model):

```
require_dependency  'project'
require_dependency  'principal'
require_dependency  'user'
```

Seems to allow standard output, please confirm.

It seems that it works fine. I will test it in more detail. In any case, this variant doesn't cause error.

Thanks!

Daniel, Thanks a lot. You soulution works for me.

I confirm it works too, but it feels like black magic to me. If anybody knows why we should declare explicitly those dependencies manually, it would be great.

Jean-Baptiste Barth wrote:

> I confirm it works too, but it feels like black magic to me. If anybody knows why we should declare explicitly those dependencies manually, it would be great.

+1

Is there a way to have something work with both Rails 2 and 3? I tried the ActionDispatch::Callbacks idiom in a Rails 2.3.14 installation and it doesn't seem to work.

What does seem to work in 2.3.14 is to use

```
config.to_prepare  do
        Monkey  Patch  things
end
```

Will this work in 3.0+? I'm just wondering if this is an idiom that might be usable in general (sorry if this is a dumb question -- I am bit hazy on the startup internals of Rails and plugins).

Actually, in fact, what about avoiding the callback entirely? If you do a:

```
require_dependency  'my_patch_file'
```

in the init.rb file and then leave your User.send by itself at the end of the patch file, shouldn't this do the right thing (i.e. loaded once in production mode and every requires in development mode)? It certainly seems to work in production mode....

Forgive me if this sounds clueless, just trying to understand options here...

For information

If I write

```
require_dependency 'project'
require_dependency 'principal'
require_dependency 'user'
```

in my plugin, the plugin
http://www.redmine.org/plugins/extended_fields
stops working, because validation

```
validates_inclusion_of :field_format, :in => Redmine::CustomFieldFormat.available_formats
```

doesn't pass.

It seems to me that the issue reported here has been overcome already; the remainder of discussion seems to resolve about a debate how to implement Monkey Patching in a way that work on both Rails 2 and 3 (although I don't understand why you would bother, RoR 3 is the future, RoR 4 is coming soon), and how to do it most elegantly etc.

So it appears this issue can be closed, can't it?

Max: not really, the underlying issue (why we need to declare dependencies explicitly) isn't resolved

John: yes, you should use @config.to_prepare { }@ in Redmine 1.x/Rails 2.x, and @ActionDispatch::Callbacks.to_prepare { }@ above. Note that since we started this thread, support for Redmine 1.x has been removed, and only Redmine 2.3.x is supported at the moment (hence Rails 3.2+).

Vladimir: that's unexpected, it should work

履歴

#1 - 2022/05/10 17:16 - Admin Redmine

- カテゴリ を Plugin API_20 にセット