

Redmine - Defect #26636

Rails 5: "Page not found" error when accessing a page of a plugin in production mode

2017-08-06 06:24 - Go MAEDA

Status:	Closed	Start date:	
Priority:	Normal	Due date:	
Assignee:	Jean-Philippe Lang	% Done:	0%
Category:	Plugin API	Estimated time:	0.00 hour
Target version:		Affected version:	
Resolution:	Fixed		

Description

Steps to reproduce:

1. Create a test plugin.

```
bin/rails g redmine_plugin foo
bin/rails g redmine_plugin_controller foo foo index
echo 'get "foo", :to => "foo#index"' >> plugins/foo/config/routes.rb
```

2. Run Redmine in production mode

```
bin/rails s -e production
```

3. Open <http://localhost/foo> and you will see "Page not found" error. No problem in development mode.

production.log:

[ShowHide](#)

Started GET "/foo" for 127.0.0.1 at 2017-08-06 13:19:10 +0900

ActionController::RoutingError (uninitialized constant FooController):

```
activesupport (5.1.2) lib/active_support/injector/methods.rb:269:in `const_get'
activesupport (5.1.2) lib/active_support/injector/methods.rb:269:in `block in constantize'
activesupport (5.1.2) lib/active_support/injector/methods.rb:267:in `each'
activesupport (5.1.2) lib/active_support/injector/methods.rb:267:in `inject'
activesupport (5.1.2) lib/active_support/injector/methods.rb:267:in `constantize'
activesupport (5.1.2) lib/active_support/dependencies.rb:582:in `get'
activesupport (5.1.2) lib/active_support/dependencies.rb:613:in `constantize'
actionpack (5.1.2) lib/action_dispatch/http/request.rb:82:in `controller_class'
actionpack (5.1.2) lib/action_dispatch/routing/route_set.rb:43:in `controller'
actionpack (5.1.2) lib/action_dispatch/routing/route_set.rb:29:in `serve'
actionpack (5.1.2) lib/action_dispatch/journey/router.rb:46:in `block in serve'
actionpack (5.1.2) lib/action_dispatch/journey/router.rb:33:in `each'
actionpack (5.1.2) lib/action_dispatch/journey/router.rb:33:in `serve'
actionpack (5.1.2) lib/action_dispatch/routing/route_set.rb:832:in `call'
rack-openid (1.4.2) lib/rack/openid.rb:98:in `call'
request_store (1.0.5) lib/request_store/middleware.rb:9:in `call'
rack (2.0.3) lib/rack/etag.rb:25:in `call'
rack (2.0.3) lib/rack/conditional_get.rb:25:in `call'
rack (2.0.3) lib/rack/head.rb:12:in `call'
rack (2.0.3) lib/rack/session/abstract/id.rb:232:in `context'
rack (2.0.3) lib/rack/session/abstract/id.rb:226:in `call'
actionpack (5.1.2) lib/action_dispatch/middleware/cookies.rb:613:in `call'
actionpack (5.1.2) lib/action_dispatch/middleware/callbacks.rb:26:in `block in call'
activesupport (5.1.2) lib/active_support/callbacks.rb:97:in `run_callbacks'
actionpack (5.1.2) lib/action_dispatch/middleware/callbacks.rb:24:in `call'
actionpack (5.1.2) lib/action_dispatch/middleware/debug_exceptions.rb:59:in `call'
actionpack (5.1.2) lib/action_dispatch/middleware/show_exceptions.rb:31:in `call'
railties (5.1.2) lib/rails/rack/logger.rb:36:in `call_app'
railties (5.1.2) lib/rails/rack/logger.rb:24:in `block in call'
```

```

activesupport (5.1.2) lib/active_support/tagged_logging.rb:69:in `block in tagged'
activesupport (5.1.2) lib/active_support/tagged_logging.rb:26:in `tagged'
activesupport (5.1.2) lib/active_support/tagged_logging.rb:69:in `tagged'
railties (5.1.2) lib/rails/rack/logger.rb:24:in `call'
actionpack (5.1.2) lib/action_dispatch/middleware/remote_ip.rb:79:in `call'
actionpack (5.1.2) lib/action_dispatch/middleware/request_id.rb:25:in `call'
rack (2.0.3) lib/rack/method_override.rb:22:in `call'
rack (2.0.3) lib/rack/runtime.rb:22:in `call'
activesupport (5.1.2) lib/active_support/cache/strategy/local_cache_middleware.rb:27:in `call'
actionpack (5.1.2) lib/action_dispatch/middleware/executor.rb:12:in `call'
actionpack (5.1.2) lib/action_dispatch/middleware/static.rb:125:in `call'
rack (2.0.3) lib/rack/content_length.rb:15:in `call'
rack (2.0.3) lib/rack/sendfile.rb:111:in `call'
railties (5.1.2) lib/rails/engine.rb:522:in `call'
puma (3.9.1) lib/puma/configuration.rb:224:in `call'
puma (3.9.1) lib/puma/server.rb:602:in `handle_request'
puma (3.9.1) lib/puma/server.rb:435:in `process_client'
puma (3.9.1) lib/puma/server.rb:299:in `block in run'
puma (3.9.1) lib/puma/thread_pool.rb:120:in `block in spawn_thread'

```

Related issues:

Related to Redmine - Feature #23630: Migrate to Rails 5.2	Closed
Related to Redmine - Patch #30725: Plugin eager_load should depend on environ...	Closed

Associated revisions

Revision 17293 - 2018-04-08 15:45 - Jean-Philippe Lang

Rails 5: "Page not found" error when accessing a page of a plugin in production mode (#26636).

Patch by Akiko Takano.

History

#1 - 2017-08-06 06:25 - Go MAEDA

- Target version set to 4.0.0

#2 - 2017-08-06 06:38 - Go MAEDA

- Description updated

#3 - 2017-08-07 06:48 - Akiko Takano

Hi, I'm trying to port my plugins, from Redmine 3.4 to Redmine trunk (ver4.0, based on Rails5).
I managed to port my plugin to support Rails5, and it seems works fine under development / test environment.

But under production mode, I saw the same error.
When I try to access one of plugin's controller, error happened and the message such as "uninitialized constant xxxxxxxController" was recorded.

I personally think, it seems required files under plugin's directory are not loaded in production mode.
And also think this is because autoload definition is disabled for production mode in Rails 5 by default.

Ref. http://edgeguides.rubyonrails.org/upgrading_ruby_on_rails.html#autoloading-is-disabled-after-booting-in-the-production-environment

Here is my workaround and I am glad if they were helpful.

Workaround 1: Enabled autoload in production mode

```

$ hg diff config/application.rb
diff -r 95623038cf82 config/application.rb
--- a/config/application.rb      Mon Jul 24 17:14:53 2017 +0000
+++ b/config/application.rb      Fri Aug 04 23:35:24 2017 +0900
@@ -12,6 +12,7 @@
 
 # Custom directories with classes and modules you want to be autoloadable.
 config.autoload_paths += %W(#{config.root}/lib)
+ config.enable_dependency_loading = true

 # Only load the plugins named here, in the order given (default is alphabetical).
 # :all can be used as a placeholder for all plugins not explicitly named.

```

Workaround 2: Add plugin's directory to eager_load_paths at lib/redmine/plugin.rb

When application boots in production then the application loads all constants found in all directories listed in eager_load_paths.

```
$ hg diff lib/redmine/plugin.rb
diff -r 95623038cf82 lib/redmine/plugin.rb
--- a/lib/redmine/plugin.rb      Mon Jul 24 17:14:53 2017 +0000
+++ b/lib/redmine/plugin.rb      Fri Aug 04 23:38:31 2017 +0900
@@ -93,6 +93,7 @@
     # Adds the app/{controllers,helpers,models} directories of the plugin to the autoload path
     Dir.glob File.expand_path(File.join(p.directory, 'app', '{controllers,helpers,models}')) do |dir|
       ActiveSupport::Dependencies.autoload_paths += [dir]
+      Rails.application.config.eager_load_paths += [dir] if Rails.env == 'production'
     end

     # Defines plugin setting if present
```

I personally prefer the second way.

#4 - 2018-04-08 15:46 - Jean-Philippe Lang

- Status changed from New to Closed
- Assignee set to Jean-Philippe Lang
- Resolution set to Fixed

Second fix committed, thanks.

#5 - 2018-04-08 21:07 - Marius BĂLTEANU

- Related to Feature #23630: Migrate to Rails 5.2 added

#6 - 2018-04-14 13:54 - Enziin System

The second way doesn't work for me.

Please use both ways.

#7 - 2018-04-15 06:58 - Go MAEDA

Enziin System wrote:

The second way doesn't work for me.

Could you describe more detail?

#8 - 2018-09-23 11:24 - Marius BĂLTEANU

- Target version deleted (4.0.0)

I'm removing this ticket from the Changelog because it was a problem generated by the migration to Rails 5 (we did the same thing with the others tickets).

#9 - 2019-02-01 02:25 - Go MAEDA

- Related to Patch #30725: Plugin eager_load should depend on environment setting instead of name added