

Redmine - Feature #32938

Rails 6: Zeitwerk support

2020-02-03 02:45 - Pavel Rosický

Status: New	Start date:
Priority: Normal	Due date:
Assignee:	% Done: 0%
Category: Rails support	Estimated time: 0.00 hour
Target version: 5.0.0	
Resolution:	
Description	
https://github.com/fxn/zeitwerk	
<p>DEPRECATION WARNING: Initialization autoloader loaded the constants ActiveRecord::Acts, ActiveRecord::Acts::Tree, Redmine::I18n, Redmine::Helpers, Redmine::Helpers::URL, Redmine::SafeAttributes, Redmine::SubclassFactory, CustomField, Redmine::Utils, Redmine::Configuration, Redmine::Scm::Adapters::CommandFailed, IssueRelation, Redmine::WikiFormatting::Macros, Redmine::Pagination, Redmine::SudoMode, ApplicationHelper, Redmine::WikiFormatting::Textile, Redmine::WikiFormatting::Textile::Formatter, Redmine::WikiFormatting::Textile::Helper, Redmine::WikiFormatting::HtmlParser, Redmine::WikiFormatting::Textile::HtmlParser, Redmine::WikiFormatting::Markdown, Redmine::WikiFormatting::Markdown::Formatter, Redmine::WikiFormatting::Markdown::HTML, Redmine::WikiFormatting::Markdown::Helper, Redmine::WikiFormatting::Markdown::HtmlParser, Redmine::Views::ApiTemplateHandler, and Setting.</p> <p>Being able to do this is deprecated. Autoloading during initialization is going to be an error condition in future versions of Rails.</p> <p>Reloading does not reboot the application, and therefore code executed during initialization does not run again. So, if you reload ActiveRecord::Acts, for example, the expected changes won't be reflected in that stale Module object.</p> <p>`config.autoloader` is set to `classic`. These autoloader constants would have been unloaded if `config.autoloader` had been set to `zeitwerk`.</p> <p>Please, check the "Autoloading and Reloading Constants" guide for solutions.</p>	
Related issues:	
Related to Redmine - Feature # 29914: Migrate to Rails 6.1	New
Duplicated by Redmine - Defect # 35004: DEPRECATION WARNING during startup	Closed

History

#1 - 2020-02-03 23:56 - Marius BALTEANU

- Related to Feature #29914: Migrate to Rails 6.1 added

#2 - 2020-03-09 23:04 - Go MAEDA

- Category set to Rails support

#3 - 2020-04-10 06:51 - Enziin System

+1 Thank you!

#4 - 2021-04-01 21:46 - Marius BALTEANU

Pavel, do you have any working patch for this?

#5 - 2021-04-02 01:10 - Pavel Rosický

sry, I don't. This is a pretty heavy change and Redmine has a lot of technical debt in this area. But I have a few hints based on my findings:

Zeitwerk depends on Rails conventions, especially the folder structure.

1/ initializers

/master/config/initializers/10-patches.rb

depends on

```
include Redmine::I18n
```

before lib/redmine.rb is loaded, so it triggers autoload. Split them into modules, require explicitly.

2/ there're multiple module definitions in a single file (many places), that don't match the filename and the expected folder structure.

3/ Ruby / Rails patches are global from the pre Ruby 2.0 era. They're patched non-deterministically in some random file

```
module ActiveRecord
  class Base
    def self.human_attribute_name(attr, options = {})
      ...
    end
  end
end
```

or a require chain like this

/master/lib/redmine/core_ext/string.rb

this issue can be avoided by using manual require at startup time, but you may consider using refinements instead

<https://www.cloudbees.com/blog/ruby-refinements/>

4/ require_dependency should be removed

/master/app/models/principal.rb#L221

here's a Rails guide and there's also an example of how to deal with STI

https://edgeguides.rubyonrails.org/autoloading_and_reloading_constants.html

5/ autoloading based on #constantize

/master/lib/redmine/wiki_formatting.rb#L42

6/ a folder structure violations, a few examples

/master/lib/diff.rb#L3

```
module RedmineDiff
  class Diff
```

```
end
end
```

should be

```
class Diff
end
```

or

/master/lib/redmine/diff.rb

```
module Redmine
  class Diff
  end
end
```

7/ module names should use camel-case, upper-case is reserved for constants

/master/lib/redmine/helpers/url.rb#L24

/master/lib/redmine/version.rb#L7

I think now it's pretty clear this is an extremely breaking change. Redmine is a part 1, but even if we fix all these issues, it's impossible to keep compatibility for plugins, but it's well, unavoidable...

Zeitwerk mode can be enabled by putting

```
config.load_defaults 6.1
```

into config/application.rb

and tested by

```
rails zeitwerk:check
```

Zeitwerk mode will become mandatory in Rails 7.

I think that decisions like module names and folder structure should be discussed with your core team. It's hard to accept a small PR without a confirmation it fixes anything and on the other hand, a big PR might include changes you won't be eventually willing to accept. I would recommend you to investigate these problems yourself, split them into smaller tasks and comment them with your expectations. The community can help you solve it this way, but this task is too big with many open questions.

#6 - 2021-04-02 01:11 - Pavel Rosický

I had to remove some references due to your spam protection :)

#7 - 2021-04-13 11:13 - Go MAEDA

- Duplicated by Defect #35004: DEPRECATION WARNING during startup added

#8 - 2021-04-13 11:13 - Go MAEDA

- Target version set to 5.0.0

#9 - 2021-04-27 09:37 - Mizuki ISHIKAWA

I haven't tried it, it seems that problem 7 can be avoided by writing config/initializers/inflations.rb.

<https://github.com/fxn/zeitwerk#inflection>