

Redmine - Feature #32938

Rails 6: Zeitwerk support

2020-02-03 02:45 - Pavel Rosický

Status:	Closed	Start date:	
Priority:	Normal	Due date:	
Assignee:	Marius BĂLTEANU	% Done:	0%
Category:	Rails support	Estimated time:	0.00 hour
Target version:			
Resolution:	Fixed		

Description

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

```
DEPRECATION WARNING: Initialization autoloaded 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.
```

Being able to do this is deprecated. Autoloading during initialization is going to be an error condition in future versions of Rails.

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.

`config.autoloader` is set to `classic`. These autoloaded constants would have been unloaded if `config.autoloader` had been set to `:zeitwerk`.

Please, check the "Autoloading and Reloading Constants" guide for solutions.

Related issues:

Related to Redmine - Feature #29914: Migrate to Rails 6.1 with Zeitwerk autol...	Closed
Related to Redmine - Defect #36218: Plugin assets are not copied correctly in...	Closed
Related to Redmine - Defect #36245: ActiveSupport::Reloader.to_prepare not wo...	Resolved
Related to Redmine - Defect #36287: rake redmine:plugins:assets fails	Closed
Related to Redmine - Defect #15958: AnonymousUser class not found by plugin	Closed
Related to Redmine - Defect #36835: Redmine::Plugin.assets_directory is not w...	Closed
Has duplicate Redmine - Defect #35004: DEPRECATION WARNING during startup	Closed

Associated revisions

Revision 21251 - 2021-10-21 00:02 - Marius BĂLTEANU

Fix rubocop "Style/CombinableLoops" offense in wiki_page.rb (#29914, #32938).

Revision 21252 - 2021-10-21 00:18 - Marius BĂLTEANU

Split multiple classes and modules that existed in the same file (#29914, #32938).

Patch by Mizuki ISHIKAWA.

Revision 21253 - 2021-10-21 00:21 - Marius BĂLTEANU

Update .rubocop_todo and fix Style/AndOr offense (#29914, #32938).

Revision 21254 - 2021-10-21 08:30 - Marius BĂLTEANU

Revert changes made in r21251 due to a test fail (#29914, #32938).

Revision 21258 - 2021-10-24 11:57 - Marius BĂLTEANU

Move custom DateValidator from lib/redmine/core_ext/active_record.rb to app/validators/date_validator.rb (#29914, #32938).

Revision 21259 - 2021-10-24 12:05 - Marius BĂLTEANU

Extract Diff and DiffTable from lib/redmine/unified_diff.rb to their own files (#29914, #32938).

Revision 21260 - 2021-10-24 12:45 - Marius BĂLTEANU

Extract Shell and DateCalculation classes from lib/redmine/utils.rb to their own files (#29914, #32938).

Revision 21261 - 2021-10-24 12:46 - Marius BĂLTEANU

Remove extra blank line (#29914, #32938).

Revision 21262 - 2021-10-27 22:42 - Marius BĂLTEANU

Move lib/diff.rb to lib/redmine/string_array_diff and extract Diffable class to its own file (#29914, #32938).

Revision 21263 - 2021-10-27 22:43 - Marius BĂLTEANU

Remove extra blank line (#29914, #32938).

Revision 21264 - 2021-10-27 22:52 - Marius BĂLTEANU

Use full full path for require (#29914, #32938).

Patch by Takashi Kato.

Revision 21265 - 2021-10-27 22:53 - Marius BĂLTEANU

Add require to acts_as_tree (#29914, #32938).

Patch by Takashi Kato.

Revision 21267 - 2021-10-28 00:27 - Marius BĂLTEANU

Rename CoreExtensions to CoreExt (#29914, #32938).

Revision 21283 - 2021-11-17 21:36 - Marius BĂLTEANU

Extract Redmine::PluginLoader from Redmine::Plugin (#32938).

Patch by Takashi Kato.

Revision 21284 - 2021-11-17 21:45 - Marius BĂLTEANU

Move preparation code to redmine/preparation.rb (#29914, #32938).

Patch by Takashi Kato.

Revision 21285 - 2021-11-17 21:47 - Marius BĂLTEANU

Add core extensions in prepare method (#29914, #32938).

Patch by Takashi Kato.

Revision 21286 - 2021-11-17 21:52 - Marius BĂLTEANU

Load core plugins using require (#29914, #32938).

Patch by Takashi Kato.

Revision 21287 - 2021-11-17 21:55 - Marius BĂLTEANU

Switch to zeitwerk autoloader (#29914, #32938).

Patch by Takashi Kato.

Revision 21288 - 2021-11-17 22:08 - Marius BĂLTEANU

Remove invalid require (#29914, #32938).

Revision 21289 - 2021-11-18 00:50 - Marius BĂLTEANU

Updates .rubocop_todo.yml (#29914, #32938).

Revision 21295 - 2021-11-25 22:34 - Marius BĂLTEANU

Fix plugin assets are no longer copied under plugin name (#36218, #29914, #32938).

Revision 21300 - 2021-12-08 02:23 - Go MAEDA

Fix redmine:plugins:assets rake task fails (#36287, #32938).

Patch by Mizuki ISHIKAWA.

History

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

- Related to Feature #29914: Migrate to Rails 6.1 with Zeitwerk autoloading 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 BĂLTEANU

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
end
```

should be

```
class Diff
end
```

or

/master/lib/redmine/diff.rb

```
module Redmine
  class Diff
    end
  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

- Has duplicate 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>

#10 - 2021-10-19 08:52 - Mizuki ISHIKAWA

- File 0001-Explicitly-require-a-class-called-in-initializers-th.patch added
- File 0002-Split-multiple-classes-and-modules-that-existed-in-t.patch added
- File 0003-Move-lib-redmine-core_ext-to-lib-core_ext.patch added
- File 0004-Remove-require_dependency.patch added
- File 0005-Fix-modules-where-folder-structure-does-not-meet-Zei.patch added

- File 0006-Change-camelcase-rules-to-keep-special-module-or-cla.patch added
- File 0007-Move-only-modules-and-classes-that-match-Zeitwerk-ru.patch added
- File 0008-Change-autoload-mode-from-classic-to-zeitwerk.patch added

I've created a draft patches to discuss Zeitwerk support.

Rails 7.0 is going to remove the old loading methods, so we need to think about how to support the new loading methods.

<https://weblog.rubyonrails.org/2021/9/15/Rails-7-0-alpha-1-released/> (Zeitwerk Exclusively section)

Description of changes. These patches have been split up as much as possible based on the description in [#32938#note-5](#).

```
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.

[0001-Explicitly-require-a-class-called-in-initializers-th.patch](#)

There are two ways to work around this, either by enclosing it in `Rails.application.reloader.to_prepare` or by explicitly writing "require" before using it. The simpler way is to enclose it in `Rails.application.reloader.to_prepare`, but the impact on the plugin is so great that we've decided to use the "require" method in 0001-Explicitly-require-a-class-called-in-initializers-th.patch.

```
Rails.application.reloader.to_prepare do
  # Autoload classes and modules needed at boot time here.
end
```

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

[0002-Split-multiple-classes-and-modules-that-existed-in-t.patch](#)

Previously, `AnonymousUser` was defined in `app/models/user.rb`, but since it is not correct according to Zeitwerk rules that the class name and file name do not match, I created a new `app/models/anonymous_user.rb` and split it. We have also made similar changes to several other classes and modules.

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/>

[0003-Move-lib-redmine-core_ext-to-lib-core_ext.patch](#)

It seemed difficult to fit this into the Zeitwerk rules, so I moved the directory.

```
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

[0004-Remove-require_dependency.patch](#)

```
5/ autoloading based on #constantize
```

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

This seems to have been resolved by other changes without a direct fix.

6/ a folder structure violations, a few examples
/master/lib/diff.rb#L3
...

[0005-Fix-modules-where-folder-structure-does-not-meet-Zei.patch](#)

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

[0006-Change-camelcase-rules-to-keep-special-module-or-cla.patch](#)

The effect of changing the class name and module name seemed to be significant, so I overwrote the movement of the camelize method without changing the name.

8 / other

[0007-Move-only-modules-and-classes-that-match-Zeitwerk-ru.patch](#)

[0008-Change-autoload-mode-from-classic-to-zeitwerk.patch](#)

When I tried to pass everything in the lib directory to autoload_paths, I found it very difficult to make the structure of the plugins and generators directories conform to Zeitwerk's rules. (Relation: 6/ a folder structure violations)
So I changed only the lib/redmine directory to lib/autoloads/redmine directory where I could apply autoloading, and set only the files under lib/autoloads directory as `config.autoload_paths += %W(#{config.root}/lib/autoloads)`.

I already know that applying these patches will cause multiple problems, mainly related to development.

- problem-1. The directory change from lib/redmine to lib/autoloads/redmine will cause many patches to fail to apply, and the changelog will be hard to read.
- problem-2. I changed redmine.rb to "require" many files directly, but autoload doesn't work for files loaded here. Even if you rewrite ApplicationHelper during development, the changes will not be reflected until you restart `rails s`.

I'm sure there are many other problems with it, so feedback and new patches would be greatly appreciated.

#11 - 2021-10-20 00:57 - Marius BĂLTEANU

Thanks Mizuki for working on this. I have as well some code, I will share it with you as soon I can.

#12 - 2021-10-21 00:25 - Marius BĂLTEANU

I've committed for now the second patch which is safe regardless the type of autoloader. Also, I've fixed some Rubocop offenses and I've updated the .rubocop_todo.yml to match the new files.

I will continue with more fixes in order to decrease the number of changes required for zeitwerk.

#13 - 2021-10-23 00:53 - Marius BĂLTEANU

- File 0001-Move-DateValidator-to-app-validators.patch added

For DateValidator, I propose to move it from lib/redmine/core_ext/active_record.rb to app/validators/date_validator.rb. It will work on both classic and zeitwerk and I think it make more sense to stay in the app. Also, other Rails projects do this. Unfortunately, I didn't find a recommended by Rails team.

#14 - 2021-10-25 01:33 - Mizuki ISHIKAWA

Marius BALTEANU wrote:

For DateValidator, I propose to move it from lib/redmine/core_ext/active_record.rb to app/validators/date_validator.rb. It will work on both classic and zeitwerk and I think it make more sense to stay in the app. Also, other Rails projects do this. Unfortunately, I didn't find a recommended by Rails team.

I also think app/validators/date_validator.rb would be better.
Thank you for your work on this issue.

#15 - 2021-10-25 08:30 - Marius BĂLTEANU

Thanks Mizuki! I've already committed the change.

In this moment, we have only one issue left that makes bin/rails zeitwerk:check to fail, is lib/diff.rb. This class is very confusing because we already have the Diff class extracted from the unified_diff.rb. I think it's better to rename the class or even better, to move it to lib/redmine/string_array_diff/

so in the end we will have:

- lib/redmine/string_array_diff/diff.rb
- lib/redmine/string_array_diff/diffable.rb

What do you think about this? Once we fix this, we can discuss the remaining deprecation warnings and hopefully we can close this issue.

Any feedback/help is appreciated.

#16 - 2021-10-25 09:52 - Mizuki ISHIKAWA

- File *split-lib-diff.patch* added

Marius BALTEANU wrote:

What do you think about this?

I think the directory structure and naming is good.

In which file do you plan to include the Diffable for Array and String? Am I correct in interpreting the changes as in the attached patch?

1. Change the module structure to match the change in directory structure. (RedmineDiff::Diff => Redmine::StringArrayDiff::Diff, RedmineDiff::Diffable => Redmine::StringArrayDiff::Diffable)
2. Keep lib/diff.rb to maintain the behavior of expanding Arrays and Strings when require 'diff' is executed in app/models/wiki_page.rb, etc.

#17 - 2021-10-26 09:32 - Marius BĂLTEANU

- File *0001-Refactor-lib-diff.rb.patch* added

Mizuki ISHIKAWA wrote:

Marius BALTEANU wrote:

What do you think about this?

I think the directory structure and naming is good.

In which file do you plan to include the Diffable for Array and String? Am I correct in interpreting the changes as in the attached patch?

1. Change the module structure to match the change in directory structure. (RedmineDiff::Diff => Redmine::StringArrayDiff::Diff, RedmineDiff::Diffable => Redmine::StringArrayDiff::Diffable)
2. Keep lib/diff.rb to maintain the behavior of expanding Arrays and Strings when require 'diff' is executed in app/models/wiki_page.rb, etc.

I think we can move everything to lib/redmine/string_array_diff and update the require in app/modeles/wiki_page.rb and lib/redmine/helpers/diff.rb as I did in attached patch. It works for you? In my CI, all tests pass.

#18 - 2021-10-26 10:15 - Mizuki ISHIKAWA

Marius BALTEANU wrote:

I think we can move everything to lib/redmine/string_array_diff and update the require in app/modeles/wiki_page.rb and lib/redmine/helpers/diff.rb as I did in attached patch. It works for you? In my CI, all tests pass.

Thanks for attaching the patch for clarity.

It looks good and actually worked fine in my environment.

#19 - 2021-10-27 11:58 - Takashi Kato

- File *alt-0001-Extract-Redmine-PluginLoader-from-Redmine-Plugin.patch* added

- File *alt-0002-Move-preparation-code-to-redmine-preparation.rb.patch* added

- File *alt-0003-Add-core-extensions-in-prepare-method.patch* added

- File *alt-0004-Load-core-plugins-with-require.patch* added

- File *alt-0005-Use-zeitwerk.patch* added

I made alternative patches to reduce require at initialization time and to increase autoloading coverage.

- I cut out Redmine::PluginLoader from Redmine::Plugin and use PluginLoader at initialization time. Now, Zeitwerk autoloading Redmine::Plugin and other classes and modules in it.
- I extracted the preparation code(building menu, setting permissions, queries) in /lib/redmine.rb to Redmine::Preparation and it's now in

Rails.application.config. to_prepare block. It reduces the use of require during initialization.

- When setting the autoloading paths for Zeitwerk, I used the namespace option. It allows that Zeitwerk autoloads libraries under the lib/redmine directory without moving them to another directory.
- To simplify patching, I've left the core plugins and RedCloth3 out of the autoload for now. I will fix them later.

#20 - 2021-10-27 12:07 - Takashi Kato

The above patches are applicable in current trunk([r21261](#)).

#21 - 2021-10-28 00:43 - Marius BĂLTEANU

Takashi Kato wrote:

The above patches are applicable in current trunk([r21261](#)).

Thanks Takashi for the patches, they look good to me. I've already committed some changes related to require from patch 0004, I've renamed CoreExtensions to CoreExt ([r21267](#)) and moved lib/diff.rb to lib/redmine/string_array_diff/diff ([r21262](#)).

Applying them on top of the changes committed today fix all the warnings related to zeitwerk:

```
⌘ redmine git:(feature/32938_zeitwerk) ⌘ bin/rails zeitwerk:check
Hold on, I am eager loading the application.
W, [2021-10-28T01:34:34.662994 #61216]  WARN -- : Creating scope :system. Overwriting existing method Enumeration.system.
W, [2021-10-28T01:34:34.858254 #61216]  WARN -- : Creating scope :sorted. Overwriting existing method Group.sorted.
W, [2021-10-28T01:34:34.929581 #61216]  WARN -- : Creating scope :sorted. Overwriting existing method User.sorted.
All is good!
```

#22 - 2021-11-01 06:57 - Mizuki ISHIKAWA

Takashi Kato wrote:

The above patches are applicable in current trunk([r21261](#)).

I've applied the patch [#32938#note-19](#) and it looks great.
I think this patch supports Zeitwerk in a better way than the patch I posted.

I've tried to run some plugins, and they work well as long as the plugin side is adapted to Zeitwerk's rules.

#23 - 2021-11-17 22:14 - Marius BĂLTEANU

- Status changed from New to Closed
- Assignee set to Marius BĂLTEANU
- Target version deleted (5.0.0)
- Resolution set to Fixed

Patches committed, thanks everyone for working on this!

#24 - 2021-11-18 00:26 - Mizuki ISHIKAWA

Thank you for your hard work.

#25 - 2021-11-25 22:35 - Marius BĂLTEANU

- Related to Defect #36218: Plugin assets are not copied correctly in trunk r21289 added

#26 - 2021-12-02 09:23 - Go MAEDA

- Related to Defect #36245: ActiveSupport::Reloader.to_prepare not working in trunk 21287 added

#27 - 2021-12-07 07:22 - Go MAEDA

- Related to Defect #36287: rake redmine:plugins:assets fails added

#28 - 2022-01-11 15:02 - Go MAEDA

- Related to Defect #15958: AnonymousUser class not found by plugin added

#29 - 2022-03-24 07:39 - Go MAEDA

- Related to Defect #36835: Redmine::Plugin.assets_directory is not working after r21283 added

#30 - 2022-03-24 07:41 - Go MAEDA

- Status changed from Closed to Reopened

Reopening this issue to handle [#36835](#).

#31 - 2022-03-25 09:56 - Marius BĂLTEANU

- Status changed from Reopened to Closed

Files

0001-Explicitly-require-a-class-called-in-initializers-th.patch	4.01 KB	2021-10-19	Mizuki ISHIKAWA
0002-Split-multiple-classes-and-modules-that-existed-in-t.patch	18.7 KB	2021-10-19	Mizuki ISHIKAWA
0003-Move-lib-redmine-core_ext-to-lib-core_ext.patch	2.29 KB	2021-10-19	Mizuki ISHIKAWA
0004-Remove-require_dependency.patch	2.06 KB	2021-10-19	Mizuki ISHIKAWA
0006-Change-camelcase-rules-to-keep-special-module-or-cla.patch	1.14 KB	2021-10-19	Mizuki ISHIKAWA
0005-Fix-modules-where-folder-structure-does-not-meet-Zei.patch	82.8 KB	2021-10-19	Mizuki ISHIKAWA
0008-Change-autoload-mode-from-classic-to-zeitwerk.patch	1.01 KB	2021-10-19	Mizuki ISHIKAWA
0007-Move-only-modules-and-classes-that-match-Zeitwerk-ru.patch	109 KB	2021-10-19	Mizuki ISHIKAWA
0001-Move-DateValidator-to-app-validators.patch	1.14 KB	2021-10-22	Marius BĂLTEANU
split-lib-diff.patch	11.6 KB	2021-10-25	Mizuki ISHIKAWA
0001-Refactor-lib-diff.rb.patch	19 KB	2021-10-26	Marius BĂLTEANU
alt-0001-Extract-Redmine-PluginLoader-from-Redmine-Plugin.patch	11 KB	2021-10-27	Takashi Kato
alt-0002-Move-preparation-code-to-redmine-preparation.rb.patch	46.4 KB	2021-10-27	Takashi Kato
alt-0003-Add-core-extensions-in-prepare-method.patch	2.4 KB	2021-10-27	Takashi Kato
alt-0004-Load-core-plugins-with-require.patch	2.67 KB	2021-10-27	Takashi Kato
alt-0005-Use-zeitwerk.patch	5.21 KB	2021-10-27	Takashi Kato