

Enable frozen string literals

2022/05/09 18:20 - Admin Redmine

ステータス:	Closed	開始日:	2022/05/09
優先度:	通常	期日:	
担当者:		進捗率:	0%
カテゴリ:	Performance_53	予定工数:	0.00時間
対象バージョン:	4.1.0_127	作業時間:	0.00時間
Redmineorg_URL:	https://www.redmine.org/issues/26561	status_id:	5
category_id:	53	tracker_id:	2
version_id:	127	plus1:	0
issue_org_id:	26561	affected_version:	
author_id:	123153	closed_on:	
assigned_to_id:	332	affected_version_id:	
comments:	23		
説明 Hi, Ruby 2.3 introduced support for frozen string literals Pros • less memory consumption • performance (less GC pressure) • safer code Cons • mutable functions like force_encoding or gsub! can't be called on frozen strings anymore, this may cause incompatibility with some plugins, but it's very easy to fix and still maintain the backwards compatibility • a magic comment has to be added at the top of each rb file :-) More information https://www.pluralsight.com/blog/software-development/ruby-2-3-working-with-immutable-strings Many popular gems and libraries already did this - for example: https://github.com/rails/rails/pull/29655 https://github.com/mikel/mail/pull/970 I attached a patch based on master that passes all tests. There're some garbage changes like removed whitespaces, I'll clean it up if it's approved. What do you think?			
journals I've added your patch to my "GitLab repository": https://gitlab.com/marius-balteanu/redmine/commits/feature/frozen_string_literals in order to run the tests and to see what is missing. Do you see any value in adding the @# frozen_string_literal: true@ to db migration files? Hi Marius, the patch is more than a year old, so there'll be some conflicts and missing files. Do you see any value in adding the # frozen_string_literal: true to db migration files?			

There was a discussion about making it default for Ruby 3, so it was historically enforced everywhere by formatters like rubocop <https://github.com/rubocop-hq/rubocop> . But in terms of performance or memory usage there probably no point for adding the comment to migration files.

one more thing, if Ruby 2.2 support is dropped #30356 instead of `"".dup` we can use `+` . That wasn't an option a year ago when even ruby 1.x was still supported.

should I update the patch and retest it on the recent redmine version?

Pavel Rosický wrote:

Hi Marius,

one more thing, if Ruby 2.2 support is dropped #30356 instead of `"".dup` we can use `+` . That wasn't an option a year ago when even ruby 1.x was still supported.

should I update the patch and retest it on the recent redmine version?

Yes, if you can do it, it will be very useful and I think we should have in mind that the support for Ruby 2.2 will be dropped in 4.1.

I'm in favour of this change, but this doesn't mean necessary that the patch will be committed. Most probably, it will be Jean-Philippe decision, but considering that almost all gems use `frozen_string_literal`, I don't see a real reason for not including it.

ok, rebased and pushed to the github
<https://github.com/redmine/redmine/pull/113>

there're many changes but most of them is just adding a magic comment

since ruby 2.2 support was dropped this should be ready for 4.1

let me know if there's anything else I can do

Since the patch is too big to review for me, I would like to take a step-by-step approach. It means that I will enable `frozen_string_literal` in each file one after another.

Today, I confirmed that we can enable `frozen_string_literal` for `test/unit/activity_test.rb` by just adding a magic comment and for `test/unit/attachment_test.rb` by applying Pavel Rosický's patch.

<https://github.com/redmine/redmine/pull/115>

this is a pull request without `frozen_string_literal` comment - 69 vs 936 changes, it should be much easier to review

you can apply the comment afterwards using rubocop like this

```
> rubocop --only Style/FrozenStringLiteralComment --auto-correct -c ruboconfig.yml
```

ruboconfig.yml

AllCops:

TargetRubyVersion: 2.3

Pavel Rosický wrote:

<https://github.com/redmine/redmine/pull/115>

this is a pull request without `frozen_string_literal` comment - 69 vs 936 changes, it should be much easier to review

Thank you for updating the patch. I would like to enable `frozen_string_literal` in files under the `./test` directory first for the following reasons:

- Changes don't affect production environments
- We can ensure that all code is executed with `frozen_string_literals` enabled by simply running `@bin/rails test@`

After that, I want to apply the rest of your patch.

Comments welcome.

Go MAEDA wrote:

Thank you for updating the patch. I would like to enable frozen_string_literal in files under the ./test directory first

I can do that, but there're some problematic methods

User#find_by_login

<https://github.com/redmine/redmine/blob/33522bf2e88026d0ca4f3d304d016dddf5966044/app/models/user.rb#L489>

Redmine::FieldFormat#parse_keyword

https://github.com/redmine/redmine/blob/4770fa5b6a12f78a5c055efc8a7bf47555496426/lib/redmine/field_format.rb#L200

and the whole codeset_utils

they're are used heavily in tests

consider this example

```
login = 'test'.force_encoding('ascii-8bit')
puts "before: #{login.encoding}" # => ascii-8bit
user = User.find_by_login(login)
puts "after: #{login.encoding}" # => utf-8
```

or in my case

```
login = 'test'.freeze
User.find_by_login(login)
=> FrozenError: can't modify frozen String
```

What is the preferred way how to deal with these issues? I would rather fix those methods to avoid side-effects, what do you think?

Pavel Rosický wrote:

What is the preferred way how to deal with these issues? I would rather fix those methods to avoid side-effects, what do you think?

You don't have to work on splitting the patch. I will do that :)

great, thanks :)

Thanks to Pavel Rosický, I have enabled frozen_string_literal for almost all files used when running the application or tests.

But I have not enabled frozen_string_literal for some files yet for the following reasons:

- lib/redmine/imap.rb and lib/redmine/pop3.rb are not covered by existing tests
- I don't have environments to test Redmine.pm or OpenId

Here is a list of files for which frozen_string_literal is still "false":

- extra/svn/reposman.rb
- extra/mail_handler/rdm-mailhandler.rb
- lib/redmine/imap.rb
- lib/redmine/pop3.rb
- lib/plugins/open_id_authentication/init.rb
- lib/plugins/open_id_authentication/lib/open_id_authentication/mem_cache_store.rb
- lib/plugins/open_id_authentication/lib/open_id_authentication/request.rb
- lib/plugins/open_id_authentication/lib/open_id_authentication/db_store.rb
- test/extra/redmine_pm/repository_git_test_pm.rb

- test/extra/redmine_pm/test_case.rb
- test/extra/redmine_pm/repository_subversion_test_pm.rb

However, I think I can close this issue because frozen_string_literal is enabled for 98% of files and features implemented by the above files are not so popular.

I hope someone tests the above files and enables frozen_string_literal for those in the future.

related_issues

relates,Closed,31004,Decode hexadecimal-encoded literals in order to be frozen string literals friendly
relates,Closed,31052,Download unified diff throws "can't modify frozen String" error.
relates,Closed,31059,Use #b shortcut instead of #force_encoding
relates,Closed,32071,FrozenError in assert_save in test/test_helper.rb
relates,Closed,31508,Add missing frozen strings and copyrights
relates,Closed,33021,[v.4.0.5-stable] Internal Server Error 500 when accessing 'repository' tab

履歴

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

- カテゴリ を Performance_53 にセット

- 対象バージョンを 4.1.0_127 にセット