Redmine - Defect #36998
Revert lazy loading of i18n files introduced in Redmine 5.0

2022-04-21 08:46 - Mischa The Evil

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Marius BALTEANU	**% Done:**	0%
Category:	I18n	**Estimated time:**	0.00 hour
Target version:	5.0.1		
Resolution:	Fixed	**Affected version:**	5.0.0

Description

Starting with 5.0.0, Redmine subclasses the ::I18n::Backend::LazyLoadable backend (#36728). I haven't looked deeply into the inner workings of this particular backend, but I did notice the following notes/recommendations in the upstream PR:

> *Most notably, a local test environment.*

> ***This backend should only be enabled in test environments!***

> *It's designed for the local test environment [...]*

Now I wonder: has the above been considered at all before it was committed? If so: what were the reasons this was nevertheless deemed fit for production use in the Redmine project?

Related issues:

Related to Redmine - Feature # 36728: Reintroduce lazy loading of i18n files	**Closed**

Associated revisions

Revision 21556 - 2022-05-02 22:28 - Marius BALTEANU

Reverts r21448 (#36998).

Revision 21557 - 2022-05-03 18:37 - Marius BALTEANU

Merge r21556 to 5.0-stable (#36998).

History

#1 - 2022-04-21 19:12 - Pavel Rosický

Hi Mischa!
thank you for catching this. Yes, there're potential thread-safety issues. Especially in production environments, everything should be eager-loaded.

by reviewing the PR again, I've found that the feature actually has to be enabled explicitly by lazy_load otherwise, it behaves the same way as SimpleBackend

I propose this patch to fix the issue, what do you think?

```
diff --git a/config/initializers/30-redmine.rb b/config/initializers/30-redmine.rb
index fc36977f6..887fcc8ee 100644
```

```
--- a/config/initializers/30-redmine.rb
+++ b/config/initializers/30-redmine.rb
@@ -4,7 +4,7 @@ require 'redmine/configuration'
 require 'redmine/plugin_loader'

 Rails.application.config.to_prepare do
-  I18n.backend = Redmine::I18n::Backend.new
+  I18n.backend = Redmine::I18n::Backend.new(lazy_load: Rails.env.development?)
   # Forces I18n to load available locales from the backend
   I18n.config.available_locales = nil
```

note that the previous change was safe, simply because the feature was disabled in all environments and this change enables it for dev environments only.

#2 - 2022-04-21 21:58 - Pavel Rosický

```
diff --git a/lib/redmine/i18n.rb b/lib/redmine/i18n.rb
index 13b84512f..c3578f34c 100644
--- a/lib/redmine/i18n.rb
+++ b/lib/redmine/i18n.rb
@@ -158,11 +158,9 @@ module Redmine
     # Custom backend based on I18n::Backend::Simple with the following changes:
     # * available_locales are determined by looking at translation file names
     class Backend < ::I18n::Backend::LazyLoadable
-      module Implementation
-        # Get available locales from the translations filenames
-        def available_locales
-          @available_locales ||= ::I18n.load_path.map {|path| File.basename(path, '.*')}.uniq.sort.map(&:to_sym)
-        end
+      # Get available locales from the translations filenames
+      def available_locales
+        @available_locales ||= ::I18n.load_path.map {|path| File.basename(path, '.*')}.uniq.sort.map(&:to_sym)
       end

       # Adds custom pluralization rules
```

#3 - 2022-04-23 10:06 - Go MAEDA

- Related to Feature #36728: Reintroduce lazy loading of i18n files added

#4 - 2022-04-27 20:23 - Holger Just

Reading a bit through the PR and the previous issues here on redmine.org, I'd rather tend to switch to the ::I18n::Backend::Simple backend again.

Within Redmine, (skipping the) load of the translations during startup doesn't seem to meaningfully affect the overall startup time. On my system, regardless of which backend or RAILS_ENV is used, full startup takes between 4 and 6 seconds with a plain Redmine.

As such, given the possible issues with the LazyLoadable backend, I think it's just not worth it and we should just use the simpler and more

deterministic ::I18n::Backend::Simple.

#5 - 2022-04-27 23:06 - Pavel Rosický

@holger even with the simple backend I18n does load translations after the first call of I18n.t, so you have to measure **the first request**, not just the startup time.

with these two patches (and lazy loading enabled), all tests passed, but I understand your concerns about complexity. Thanks for considering it!

#6 - 2022-04-27 23:21 - Marius BALTEANU

- Target version set to 5.0.1

Is it enough to revert r21448 or we should allow lazy loading on dev environment?

#7 - 2022-04-28 13:04 - Holger Just

I'd rather just revert r21448 unless someone can show a real tangible upside of this diversion between environments (which I don't see right now).

#8 - 2022-04-28 16:45 - Pavel Rosický

these diversions between environments are common - auto_load in development vs eager_load in production

but if you think the risk is too high, please revert r21448 thank you.

#9 - 2022-05-02 22:30 - Marius BALTEANU

- Resolution set to Fixed

Reverted r21448 in r21556.

#10 - 2022-05-02 22:30 - Marius BALTEANU

- Status changed from New to Resolved
- Assignee set to Marius BALTEANU

#11 - 2022-05-03 18:41 - Marius BALTEANU

- Subject changed from ::I18n::Backend::LazyLoadable is not deemed fit for production use to Revert lazy loading of i18n files introduced in Redmine 5.0
- Status changed from Resolved to Closed

Reverted in r21556 and merged to 5.0-stable branch. Thank you all for your reviews!

Pavel, I agree with Holger, we can discuss this again for non production environments if we can obtain a meaningfully improvement.

#12 - 2022-05-03 19:22 - Pavel Rosický

note that Redmine used to have its own backend based on the same idea for years (originally introduced by JPLang in 2012 :)), so this isn't something new
https://github.com/redmine/redmine/commit/3739810afa3545e6747a9111185dc8808fff6078

but I agree it's better to keep it as simple as possible. Thank you all for your comments!