Redmine - Defect #37249

Missing rexml gem causes errors in PUT - Adding the gem manually everything works

2022-06-14 17:17 - Bubi =

Status:	Closed		**Start date:**	
Priority:	Normal		**Due date:**	
Assignee:	Marius BĂLTEANU		**% Done:**	0%
Category:	Gems support		**Estimated time:**	0.00 hour
Target version:	5.0.2			
Resolution:	Fixed		**Affected version:**	5.0.1

Description

Here the part of the log

```
App 19843 output: [599f0f62-257a-49a9-bf49-2d9cd67a9566] Started POST "/issues.xml?key=hidden" for
 hidden at 2022-06-14 14:47:42 +0000
App 19843 output: You don't have rexml installed in your application. Please add it to your Gemfil
e and run bundle install
App 19843 output: [599f0f62-257a-49a9-bf49-2d9cd67a9566] LoadError (cannot load such file -- rexml
/document):
App 19843 output: [599f0f62-257a-49a9-bf49-2d9cd67a9566] zeitwerk (2.6.0) lib/zeitwerk/kernel.rb:3
5:in `require'
```

Related issues:

Related to Redmine - Feature #34992: Ruby 3.0 support	**Closed**
Related to Redmine - Patch #37258: Switch default backend of ActiveSupport::X...	**Closed**

Associated revisions

Revision 21635 - 2022-06-16 17:04 - Marius BĂLTEANU

Require rexml gem for ruby >= 3.0 (#37249).

Patch by Holger Just.

Revision 21636 - 2022-06-16 17:05 - Marius BĂLTEANU

Merged r21635 to 5.0-stable (#37249).

History

#1 - 2022-06-14 21:10 - Holger Just

- Status changed from New to Confirmed

rexml used to be a a requirement of Rails as it is used as the default backend of ActiveSupport::XmlMini. This backend is used e.g. when parsing XML payloads of API requests.

In Rails 6.1 the dependency to rexml was removed but the default backend specification still remains.

In Ruby <= 2.7, the rexml library was included in the Ruby standard library. As such, it was still present and available even when not directly required. Starting with Ruby 3.0 however, rexml was converted into a bundled gem. As such, it is necessary to define it as a dependency in bundler (either explicitly or transitive via other dependencies). Now, if Redmine is installed with its test group (which is the default), rexml is installed as transitive dependency to rubocop and is thus available.

However, if bundler is configured with

```
bundle config --without test
```

on Ruby >= 3.0, rexml is not installed and is hidden by bundler, even if it's present as a bundled gem in Ruby. This results in ActiveSupport::XmlMini trying to load rexml fir its default backkend and failing.

We have two ways to resolve this:

1. we could add a strict rexml dependency to the Gemfile, effectively re-introducing the previous rails dependency:

```
gem "rexml", require: false
```

2. Alternatively, we could switch to a different backend. As Redmine already depends on nokogiri, we could thus switch the XmlMini backend to nokogiri in an initializer, e.g. config/initializers/30-redmine.rb:

```
ActiveSupport::XmlMini.backend = 'Nokogiri'
```

This likely has the added benefit of (much) faster XML parsing and generation.

#2 - 2022-06-15 11:27 - Go MAEDA

Holger Just wrote:

1. we could add a strict rexml dependency to the Gemfile, effectively re-introducing the previous rails dependency:
   [...]
2. Alternatively, we could switch to a different backend. As Redmine already depends on nokogiri, we could thus switch the XmlMini backend to nokogiri in an initializer, e.g. config/initializers/30-redmine.rb:
   [...]
   This likely has the added benefit of (much) faster XML parsing and generation.

I would prefer the first approach. This is because the fix should be backported to 5.0-stable branch and delivered in 5.0.2, but I think switching a backend to Nokogiri is a somewhat big change for the maintenance release.

#3 - 2022-06-15 11:27 - Go MAEDA

- *Target version set to 5.0.2*

#4 - 2022-06-15 11:29 - Go MAEDA

- *Related to Feature #34992: Ruby 3.0 support added*

#5 - 2022-06-16 16:51 - Holger Just

Go MAEDA wrote:

I would prefer the first approach. This is because the fix should be backported to 5.0-stable branch and delivered in 5.0.2, but I think switching a backend to Nokogiri is a somewhat big change for the maintenance release.

In that case, I would be in favor of adding the rexml gem as a dependency for Ruby >= 3.0 on 5.0-stable only, i.e. to add the following patch in 5.0-stable:

```
diff --git a/Gemfile b/Gemfile
index cf7a8d78fa..29d639b9a7 100644
--- a/Gemfile
+++ b/Gemfile
@@ -14,6 +14,7 @@ gem 'marcel'
 gem "mail", "~> 2.7.1"
 gem 'csv', '~> 3.2.0'
 gem 'nokogiri', (Gem.ruby_version < Gem::Version.new('2.6.0') ? '~> 1.12.5' : '~> 1.13.4')
+gem "rexml", require: false if Gem.ruby_version >= Gem::Version.new('3.0')
 gem 'i18n', '~> 1.10.0'
 gem "rbpdf", "~> 1.20.0"
 gem 'addressable'
```

For trunk, we should instead opt to set the XmlMini backend to "Nokogiri" and get rid of rexml entirely. This avoids the usage of multiple different XML engines and the associated potential security issues due to different update-policies here.

#6 - 2022-06-16 17:06 - Marius BĂLTEANU

- *Status changed from Confirmed to Resolved*

- *Assignee set to Marius BĂLTEANU*

- *Resolution set to Fixed*

Thanks for reporting and fixing the issue. I've fixed this in trunk and 5.0-stable by requiring the gem.

Holger, I'm going to open a new issue to handle Nokogiri backend for 5.1.0.

#7 - 2022-06-16 23:25 - Marius BĂLTEANU

- *Precedes Patch #37258: Switch default backend of ActiveSupport::XmlMini from rexml to Nokogiri added*

#8 - 2022-06-16 23:25 - Marius BĂLTEANU

- Precedes deleted (Patch #37258: Switch default backend of ActiveSupport::XmlMini from rexml to Nokogiri)

#9 - 2022-06-16 23:25 - Marius BĂLTEANU

- Related to Patch #37258: Switch default backend of ActiveSupport::XmlMini from rexml to Nokogiri added

#10 - 2022-06-16 23:26 - Marius BĂLTEANU

- Status changed from Resolved to Closed

I've opened [#37258](#).

#11 - 2022-06-16 23:26 - Marius BĂLTEANU

- Category changed from REST API to Gems support