

Redmine - Defect #31387

Don't rescue Exception class

2019-05-17 20:51 - Pavel Rosický

Status:	Closed	Start date:	
Priority:	Normal	Due date:	
Assignee:	Go MAEDA	% Done:	0%
Category:	Code cleanup/refactoring	Estimated time:	0.00 hour
Target version:	4.1.0	Affected version:	
Resolution:	Fixed		
Description			
https://stackoverflow.com/questions/10048173/why-is-it-bad-style-to-rescue-exception-e-in-ruby			
https://thoughtbot.com/blog/rescue-standarderror-not-exception			
Exception is the root of Ruby's exception hierarchy, so when you rescue Exception you rescue from everything, including subclasses such as SyntaxError, LoadError, and Interrupt.			
Rescuing Interrupt prevents the user from using CTRL-C to exit the program.			
I've seen a recent addition to redmine			
<pre>def destroy IssueStatus.find(params[:id]).destroy redirect_to issue_statuses_path rescue Exception => e flash[:error] = I(:error_unable_delete_issue_status, ERB::Util.h(e.message)) redirect_to issue_statuses_path end</pre>			
https://github.com/redmine/redmine/blob/aed2d197a0e897e8a83013384b95353e110bd2f8/app/controllers/issue_statuses_controller.rb#L77			
please replace these "safe" exceptions. StandardError is usually more than sufficient.			
Related issues:			
Related to Redmine - Feature #31361: Include a reason in the error message wh...		Closed	
Related to Redmine - Patch #29441: Remove code related to JRuby and unsupport...		Closed	

Associated revisions

Revision 18187 - 2019-05-22 01:53 - Go MAEDA

Custom Exception classes should inherit StandardError instead of Exception (#31387).

Patch by Go MAEDA.

Revision 18197 - 2019-05-25 08:50 - Go MAEDA

Don't rescue Exception class (#31387).

Patch by Go MAEDA and Pavel Rosický.

History

#1 - 2019-05-18 06:55 - Go MAEDA

Thank you for pointing it out.

Maybe we should also check the following lines, shouldn't we?

```
$ egrep -rn 'rescue\s+Exception' app config lib extra
app/models/mailer.rb:705: rescue Exception => e
```

```

app/models/repository/git.rb:87: rescue Exception => e
app/models/repository.rb:385: rescue Exception => e
app/models/repository.rb:395: rescue Exception => e
app/models/repository.rb:405: rescue Exception => e
app/models/mail_handler.rb:56: rescue Exception => e
app/models/import.rb:58: rescue Exception => e
app/models/import.rb:262: rescue Exception => e
app/controllers/auth_sources_controller.rb:63: rescue Exception => e
app/controllers/admin_controller.rb:57: rescue Exception => e
app/controllers/admin_controller.rb:68: rescue Exception => e
config/routes.rb:358: rescue Exception => e
lib/redmine/plugin.rb:421: rescue Exception => e
lib/redmine/plugin.rb:432: rescue Exception => e
lib/redmine/plugin.rb:443: rescue Exception => e
lib/redmine/scm/adapters/mercurial_adapter.rb:95: rescue Exception => e
lib/redmine/scm/adapters/subversion_adapter.rb:118: rescue Exception => e
lib/redmine/scm/adapters/abstract_adapter.rb:254: rescue Exception => e
lib/redmine/scm/adapters/abstract_adapter.rb:285: rescue Exception => err
lib/tasks/email.rake:167: rescue Exception => e
lib/tasks/migrate_from_trac.rake:617: rescue Exception => e
lib/tasks/migrate_from_trac.rake:632: rescue Exception => e
lib/tasks/locales.rake:171: rescue Exception => e1
lib/tasks/locales.rake:176: rescue Exception => e
extra/svn/reposman.rb:57: rescue Exception => e
extra/mail_handler/rdm-mailhandler.rb:207: rescue Exception => e

```

#2 - 2019-05-20 12:01 - Go MAEDA

- Related to Feature #31361: Include a reason in the error message when an issue status cannot be deleted added

#3 - 2019-05-20 13:46 - Go MAEDA

- File 0001-Custom-Exception-classes-should-inherit-StandardError.patch added

- File 0002-Don-t-rescue-Exception.patch added

- Category set to Code cleanup/refactoring

- Target version set to Candidate for next minor release

Here are patches to fix not to raise/rescue Exception. I hope someone to review these patches.

#4 - 2019-05-20 14:09 - Pavel Rosický

thanks, there's a typo in config/routes.rb

it should be rescue SyntaxError, StandardError => e

#5 - 2019-05-20 14:26 - Go MAEDA

- File 0002-Don-t-rescue-Exception.patch added

Pavel Rosický wrote:

it should be rescue SyntaxError, StandardError => e

Thanks, updated the patch. Could you tell me why it catches SyntaxError? To detect syntax error in plugins' routes.rb?

#6 - 2019-05-20 14:26 - Go MAEDA

- File deleted (0002-Don-t-rescue-Exception.patch)

#7 - 2019-05-20 14:59 - Pavel Rosický

exceptions have a hierarchy

```

Exception
  NoMemoryError
  ScriptError
    LoadError
    NotImplementedError
    SyntaxError
  SignalException

```

```
Interrupt
StandardError
  ArgumentError
  IOError
  EOFError
  IndexError
  LocalJumpError
  NameError
  NoMethodError
  RangeError
  FloatDomainError
  RegexpError
  RuntimeError
  SecurityError
  SystemCallError
  SystemStackError
  ThreadError
  TypeError
  ZeroDivisionError
SystemExit
```

as you can see `SyntaxError` doesn't inherit from `StandardError`. It's a common mistake and the original intention was to show a better message that there's an error in a plugin instead of a long and confusing backtrace that Ruby would usually print.

#8 - 2019-05-21 05:33 - Go MAEDA

- Target version changed from Candidate for next minor release to 4.1.0

Setting the target version to 4.1.0.

#9 - 2019-05-21 16:41 - Go MAEDA

- Assignee set to Go MAEDA

#10 - 2019-05-22 09:59 - Go MAEDA

Pavel Rosický wrote:

as you can see `SyntaxError` doesn't inherit from `StandardError`. It's a common mistake and the original intention was to show a better message that there's an error in a plugin instead of a long and confusing backtrace that Ruby would usually print.

You are right.

With catching `SyntaxError`:

```
An error occurred while loading the routes definition of redmine_message_customize plugin (/Users/maeda/redmines/redmine-trunk/plugins/redmine_message_customize/config/routes.rb): (eval):7: syntax error, unexpected end-of-input, expecting '}'.
```

Without catching `SyntaxError`:

[ShowHide](#)

```
Traceback (most recent call last):
 47: from bin/rails:4:in `'
 46: from bin/rails:4:in `require'
 45: from /Users/maeda/redmines/gems/ruby/2.6.0/gems/railties-5.2.3/lib/rails/commands.rb:18:in `'
 44: from /Users/maeda/redmines/gems/ruby/2.6.0/gems/railties-5.2.3/lib/rails/command.rb:46:in `invoke'
 43: from /Users/maeda/redmines/gems/ruby/2.6.0/gems/railties-5.2.3/lib/rails/command/base.rb:65:in `perform'
 42: from /Users/maeda/redmines/gems/ruby/2.6.0/gems/thor-0.20.3/lib/thor.rb:387:in `dispatch'
 41: from /Users/maeda/redmines/gems/ruby/2.6.0/gems/thor-0.20.3/lib/thor/invoke.rb:126:in `invoke_command'
 40: from /Users/maeda/redmines/gems/ruby/2.6.0/gems/thor-0.20.3/lib/thor/command.rb:27:in `run'
 39: from /Users/maeda/redmines/gems/ruby/2.6.0/gems/railties-5.2.3/lib/rails/commands/console/console_command.rb:95:in `perform'
 38: from /Users/maeda/redmines/gems/ruby/2.6.0/gems/railties-5.2.3/lib/rails/command/actions.rb:15:in `require_application_and_environment!'
 37: from /Users/maeda/redmines/gems/ruby/2.6.0/gems/railties-5.2.3/lib/rails/command/actions.rb:28:in `require_environment!'
 36: from /Users/maeda/redmines/gems/ruby/2.6.0/gems/railties-5.2.3/lib/rails/application.rb:337:in `require_environment!'
```

```

35: from /Users/maeda/redmines/gems/ruby/2.6.0/gems/railties-5.2.3/lib/rails/application.rb:337:in `require'
34: from /Users/maeda/redmines/redmine-trunk/config/environment.rb:16:in `<top (required)>'
33: from /Users/maeda/redmines/gems/ruby/2.6.0/gems/railties-5.2.3/lib/rails/application.rb:361:in `initialize!'
32: from /Users/maeda/redmines/gems/ruby/2.6.0/gems/railties-5.2.3/lib/rails/initializable.rb:60:in `run_initializers'
31: from /Users/maeda/.rbenv/versions/2.6.1/lib/ruby/2.6.0/tsort.rb:205:in `tsort_each'
30: from /Users/maeda/.rbenv/versions/2.6.1/lib/ruby/2.6.0/tsort.rb:226:in `tsort_each'
29: from /Users/maeda/.rbenv/versions/2.6.1/lib/ruby/2.6.0/tsort.rb:347:in `each_strongly_connected_component'
28: from /Users/maeda/.rbenv/versions/2.6.1/lib/ruby/2.6.0/tsort.rb:347:in `call'
27: from /Users/maeda/.rbenv/versions/2.6.1/lib/ruby/2.6.0/tsort.rb:347:in `each'
26: from /Users/maeda/.rbenv/versions/2.6.1/lib/ruby/2.6.0/tsort.rb:349:in `block in each_strongly_connected_component'
25: from /Users/maeda/.rbenv/versions/2.6.1/lib/ruby/2.6.0/tsort.rb:431:in `each_strongly_connected_component_from'
24: from /Users/maeda/.rbenv/versions/2.6.1/lib/ruby/2.6.0/tsort.rb:350:in `block (2 levels) in each_strongly_connected_component'
23: from /Users/maeda/.rbenv/versions/2.6.1/lib/ruby/2.6.0/tsort.rb:228:in `block in tsort_each'
22: from /Users/maeda/redmines/gems/ruby/2.6.0/gems/railties-5.2.3/lib/rails/initializable.rb:61:in `block in run_initializers'
21: from /Users/maeda/redmines/gems/ruby/2.6.0/gems/railties-5.2.3/lib/rails/initializable.rb:32:in `run'
20: from /Users/maeda/redmines/gems/ruby/2.6.0/gems/railties-5.2.3/lib/rails/initializable.rb:32:in `instance_exec'
19: from /Users/maeda/redmines/gems/ruby/2.6.0/gems/railties-5.2.3/lib/rails/application/finisher.rb:130:in `block in <module:Finisher>'
18: from /Users/maeda/redmines/gems/ruby/2.6.0/gems/railties-5.2.3/lib/rails/application/routes_reloader.rb:10:in `execute'
17: from /Users/maeda/redmines/gems/ruby/2.6.0/gems/activesupport-5.2.3/lib/active_support/file_update_checker.rb:83:in `execute'
16: from /Users/maeda/redmines/gems/ruby/2.6.0/gems/railties-5.2.3/lib/rails/application/routes_reloader.rb:30:in `block in updater'
15: from /Users/maeda/redmines/gems/ruby/2.6.0/gems/railties-5.2.3/lib/rails/application/routes_reloader.rb:20:in `reload!'
14: from /Users/maeda/redmines/gems/ruby/2.6.0/gems/railties-5.2.3/lib/rails/application/routes_reloader.rb:41:in `load_paths'
13: from /Users/maeda/redmines/gems/ruby/2.6.0/gems/railties-5.2.3/lib/rails/application/routes_reloader.rb:41:in `each'
12: from /Users/maeda/redmines/gems/ruby/2.6.0/gems/railties-5.2.3/lib/rails/application/routes_reloader.rb:41:in `block in load_paths'
11: from /Users/maeda/redmines/gems/ruby/2.6.0/gems/activesupport-5.2.3/lib/active_support/dependencies.rb:285:in `load'
10: from /Users/maeda/redmines/gems/ruby/2.6.0/gems/activesupport-5.2.3/lib/active_support/dependencies.rb:257:in `load_dependency'
9: from /Users/maeda/redmines/gems/ruby/2.6.0/gems/activesupport-5.2.3/lib/active_support/dependencies.rb:285:in `block in load'
8: from /Users/maeda/redmines/gems/ruby/2.6.0/gems/activesupport-5.2.3/lib/active_support/dependencies.rb:285:in `load'
7: from /Users/maeda/redmines/redmine-trunk/config/routes.rb:20:in `<top (required)>'
6: from /Users/maeda/redmines/gems/ruby/2.6.0/gems/actionpack-5.2.3/lib/action_dispatch/routing/route_set.rb:414:in `draw'
5: from /Users/maeda/redmines/gems/ruby/2.6.0/gems/actionpack-5.2.3/lib/action_dispatch/routing/route_set.rb:432:in `eval_block'
4: from /Users/maeda/redmines/gems/ruby/2.6.0/gems/actionpack-5.2.3/lib/action_dispatch/routing/route_set.rb:432:in `instance_exec'
3: from /Users/maeda/redmines/redmine-trunk/config/routes.rb:355:in `block in <top (required)>'
2: from /Users/maeda/redmines/redmine-trunk/config/routes.rb:355:in `glob'
1: from /Users/maeda/redmines/redmine-trunk/config/routes.rb:359:in `block (2 levels) in <top (required)>'
/Users/maeda/redmines/redmine-trunk/config/routes.rb:359:in `instance_eval': (eval):7: syntax error, unexpected end-of-input, expecting '}' (SyntaxError)

```

#11 - 2019-05-25 03:05 - Go MAEDA

According to [r6230](#), it seems that we have to catch Exception in lib/redmine/scm/adapters/abstract_adapter.rb unless we drop the support for JRuby.

```

Index: lib/redmine/scm/adapters/abstract_adapter.rb
=====
--- lib/redmine/scm/adapters/abstract_adapter.rb      (000000 6229)
+++ lib/redmine/scm/adapters/abstract_adapter.rb      (000000 6230)
@@ -224,7 +224,11 @@
     io.close_write

```

```

        block.call(io) if block_given?
      end
-     rescue Errno::ENOENT => e
+     ## If scm command does not exist,
+     ## Linux JRuby 1.6.2 (ruby-1.8.7-p330) raises java.io.IOException
+     ## in production environment.
+     # rescue Errno::ENOENT => e
+     rescue Exception => e
        msg = strip_credential(e.message)
        # The command failed, log it and re-raise
        logmsg = "SCM command failed, "

```

#12 - 2019-05-25 03:08 - Go MAEDA

- Related to Patch #29441: Remove code related to JRuby and unsupported Ruby versions added

#13 - 2019-05-25 07:42 - Pavel Rosický

This was a fix for a very, very old jruby version. It should match MRI's behaviour now.

#14 - 2019-05-25 08:53 - Go MAEDA

- Subject changed from don't rescue Exception to Don't rescue Exception class

- Status changed from New to Closed

- Resolution set to Fixed

Committed the patch.

Pavel Rosický wrote:

This was a fix for a very, very old jruby version. It should match MRI's behaviour now.

Thank you for the advice. I committed the following code.

```

Index: lib/redmine/scm/adapters/abstract_adapter.rb
=====
--- lib/redmine/scm/adapters/abstract_adapter.rb      (000000 18196)
+++ lib/redmine/scm/adapters/abstract_adapter.rb      (000000 18197)
@@ -247,11 +247,7 @@
         io.close_write unless options[:write_stdin]
         block.call(io) if block_given?
       end
-     ## If scm command does not exist,
-     ## Linux JRuby 1.6.2 (ruby-1.8.7-p330) raises java.io.IOException
-     ## in production environment.
-     # rescue Errno::ENOENT => e
-     rescue Exception => e
+     rescue => e
        msg = strip_credential(e.message)
        # The command failed, log it and re-raise
        logmsg = "SCM command failed, "

```

Files

File Name	Size	Date	Author
0001-Custom-Exception-classes-should-inherit-StandardError.patch	3.78 KB	2019-05-20	Go MAEDA
0002-Don-t-rescue-Exception.patch	19.3 KB	2019-05-20	Go MAEDA