Redmine - Defect #37476

Psych::DisallowedClass exception when loading default plugin settings

2022-07-20 14:48 - Dmitry Makurin

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Go MAEDA	**% Done:**	0%
Category:	Plugin API	**Estimated time:**	0.00 hour
Target version:			
Resolution:	Fixed	**Affected version:**	5.0.2

Description

After r21722 redmine keeps throwing a Psych::DisallowedClass (Tried to load unspecified class: Symbol) when accessing default plugin setting using Setting.plugin_name.

Turns out settings for plugins are loading either from database (which is HashWithIndifferentAccess) and if it is not found then fallbacks to init.rb where it's a simple Hash object. Passing latter to YAML.safe_load raises a error.

To reproduce install any plugin that defines default settings in init.rb and call Setting.plugin_name.

Here for example I installed redmine_issue_templates and then in console trying to get its settings:

```
$ bundle exec rails db:reset RAILS_ENV=development
Dropped database 'db/development.sqlite3'
Created database 'db/development.sqlite3'
makurin@makurin:~/Work/clean-redmine$ rails c
Loading development environment (Rails 6.1.6.1)
irb(main):001:0> Setting.plugin_redmine_issue_templates
   (2.4ms)  SELECT sqlite_version(*)
  Setting Load (0.4ms)  SELECT "settings".* FROM "settings" WHERE "settings"."name" = ? ORDER BY "
settings"."id" DESC LIMIT ?  [["name", "plugin_redmine_issue_templates"], ["LIMIT", 1]]
Traceback (most recent call last):
        3: from app/models/setting.rb:320:in `plugin_redmine_issue_templates'
        2: from app/models/setting.rb:125:in `[]'
        1: from app/models/setting.rb:111:in `value'
Psych::DisallowedClass (Tried to load unspecified class: Symbol)
```

So far I found two options to resolve it.

Tell YAML that Symbol is allowed:

```
diff --git a/app/models/setting.rb b/app/models/setting.rb
index 53b88bcad4..bc59306476 100644
--- a/app/models/setting.rb
+++ b/app/models/setting.rb
@@ -108,7 +108,7 @@ class Setting < ActiveRecord::Base
     v = read_attribute(:value)
     # Unserialize serialized settings
     if available_settings[name]['serialized'] && v.is_a?(String)
-      v = YAML.safe_load(v, permitted_classes: [ActiveSupport::HashWithIndifferentAccess])
+      v = YAML.safe_load(v, permitted_classes: [ActiveSupport::HashWithIndifferentAccess, Symbol]
)
       v = force_utf8_strings(v)
     end
     v = v.to_sym if available_settings[name]['format'] == 'symbol' && !v.blank?
```

Or convert default plugin settings to HashWithIndifferentAccess:

```
diff --git a/app/models/setting.rb b/app/models/setting.rb
index 53b88bcad4..a5f7b7966c 100644
--- a/app/models/setting.rb
+++ b/app/models/setting.rb
@@ -293,7 +293,7 @@ class Setting < ActiveRecord::Base
```

```
   def self.define_plugin_setting(plugin)
     if plugin.settings
       name = "plugin_#{plugin.id}"
-      define_setting name, {'default' => plugin.settings[:default], 'serialized' => true}
+      define_setting name, {'default' => plugin.settings[:default].with_indifferent_access, 'seri
alized' => true}
     end
   end
```

Associated revisions

Revision 21725 - 2022-07-21 02:41 - Go MAEDA

Psych::DisallowedClass exception when loading default plugin settings (#37450, #37476).

Contributed by Dmitry Makurin.

Revision 21729 - 2022-07-21 03:41 - Go MAEDA

Merged r21725 from trunk to 5.0-stable (#37450, #37476).

Revision 21730 - 2022-07-21 04:52 - Go MAEDA

Merged r21725 from trunk to 4.2-stable (#37450, #37476).

Revision 21777 - 2022-08-25 15:47 - Go MAEDA

Use the classes whitelist configured in application.rb instead of hardcoded classes (#37476).

Patch by Jens Krämer.

Revision 21778 - 2022-08-25 15:53 - Go MAEDA

Merged r21777 from trunk to 5.0-stable (#37476).

Revision 21779 - 2022-08-25 15:54 - Go MAEDA

Merged r21777 from trunk to 4.2-stable (#37476).

History

#1 - 2022-07-20 15:07 - Go MAEDA

- Status changed from New to Confirmed

#3 - 2022-07-21 00:01 - Go MAEDA

The following test can detect the reported issue.

```
diff --git a/test/unit/lib/redmine/plugin_test.rb b/test/unit/lib/redmine/plugin_test.rb
index 54394ab57..a5a1b2aa3 100644
--- a/test/unit/lib/redmine/plugin_test.rb
+++ b/test/unit/lib/redmine/plugin_test.rb
@@ -196,6 +196,13 @@ class Redmine::PluginTest < ActiveSupport::TestCase
     end
   end

+  def test_default_settings
+    @klass.register(:foo_plugin) {settings :default => {'key1' => 'abc', :key2 => 123}}
+    h = Setting.plugin_foo_plugin
+    assert_equal 'abc', h['key1']
+    assert_equal 123, h[:key2]
+  end
+
   def test_settings_warns_about_possible_partial_collision
     @klass.register(:foo_plugin) {settings :partial => 'foo/settings'}
     Rails.logger.expects(:warn)

Error:
Redmine::PluginTest#test_default_settings:
Psych::DisallowedClass: Tried to load unspecified class: Symbol
    app/models/setting.rb:111:in `value'
    app/models/setting.rb:125:in `[]'
    app/models/setting.rb:320:in `plugin_foo_plugin'
    test/unit/lib/redmine/plugin_test.rb:201:in `test_default_settings'
```

```
rails test test/unit/lib/redmine/plugin_test.rb:199
```

#4 - 2022-07-21 04:52 - Go MAEDA

- Subject changed from Error when loading default plugin settings to Psych::DisallowedClass exception when loading default plugin settings

- Category set to Plugin API

- Status changed from Confirmed to Closed

- Assignee set to Go MAEDA

- Resolution set to Fixed

Committed the fix. Thank you for quickly reporting the issue.

#5 - 2022-08-25 13:07 - Jens Krämer

For consistency reasons, I would suggest to use the whitelist configured for ActiveRecord serialization in application.rb in setting.rb as well:

```
diff --git a/app/models/setting.rb b/app/models/setting.rb
index 748d6543ac..3d1048d7b1 100644
--- a/app/models/setting.rb
+++ b/app/models/setting.rb
@@ -119,7 +119,7 @@ class Setting < ActiveRecord::Base
     v = read_attribute(:value)
     # Unserialize serialized settings
     if available_settings[name]['serialized'] && v.is_a?(String)
-      v = YAML.safe_load(v, permitted_classes: [Symbol, ActiveSupport::HashWithIndifferentAccess])
+      v = YAML.safe_load(v, permitted_classes: Rails.configuration.active_record.yaml_column_permitted_classe
s)
       v = force_utf8_strings(v)
     end
     v = v.to_sym if available_settings[name]['format'] == 'symbol' && !v.blank?
```

Some plugins do serialize other classes, this way only a single list of exceptions needs to be maintained.

#6 - 2022-08-25 15:55 - Go MAEDA

Jens Krämer wrote:

> For consistency reasons, I would suggest to use the whitelist configured for ActiveRecord serialization in application.rb in setting.rb as well:
>
> [...]
>
> Some plugins do serialize other classes, this way only a single list of exceptions needs to be maintained.

Thank you for suggesting the fix. I have committed the fix.