Redmine - Feature #27705

Gemify redmine plugins

2017-12-02 13:41 - Sho HASHIMOTO

Status:	New	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:	Plugin API	**Estimated time:**	0.00 hour
Target version:			
Resolution:			

Description

I want to install redmine plugins by adding gem name to Gemfile.local like redmine_github_hook plugin.

Current installing/updating method have below cautions for me.

- Difficult to specify plugin version
  - Server maintainer should exec `$ git clone [--depth=1] -b <release version> https://github.com/foo/bar/`, but plugin README often describes installing master
- Difficult to detect plugins vulnerability with using bundler-audit
  - We have to check each plugins by hand now

I'm happy if Redmine::Plugin.load requires gem's init.rb.

- Pros
  - Easy to install plugin: only writing gem name to Gemfile.local
  - Easy to specify plugin version: only specifying gem version to Gemfile.local
  - Easy to update plugin: only executing `$ bundle update <plugin gem name>`
  - Easy to detect **(only gemified)** plugins vulnerability with using bundler-audit
- Cons
  - Redmine should detect wheather redmine plugin or not from all bundle installed gem

For example, support_gem_plugin.patch can load gemified plugin.

Gemfile.local example is below(I fixed little existing sidebar_hide plugin).

```
gem 'sidebar_hide', github: 'sho-h/sidebar_hide', branch: 'redmine_support_gem_plugin_test'
```

This was gemified and specified directory in init.rb.

History

#1 - 2017-12-31 17:43 - Toshi MARUYAMA

+100

I have a question.
"require initializer" is called twice. Is it correct?
One is in this patch and another is in original code.

#2 - 2018-01-04 16:16 - Sho HASHIMOTO

> +100

Thanks!

> "require initializer" is called twice. Is it correct?
> One is in this patch and another is in original code.

Oh... Maybe it's cause problem.

I left original code. Original code is for loading current structure plugins.

I think, there are below patterns.

1. Same plugin and same version: Maybe, problem doesn't break to the surface.
2. Same plugin and different version: Cause problems. Both require will success. Overwritten by plugin dir's one halfway.
3. Different plugins and same ID: Cause problems. Both require will success. Each function will not be conflicted. But redmine will misunderstand only plugin dir's one was loaded.
4. Different plugins and different IDs: No problem.

1., 2., 3. have same problem. @registered_plugins[id] was overwritten.
To fix this, for example, Redmine::Plugin.register(id) raises(or fails) if plugin was already installed.

```
 def self.register(id, &block)
+  raise ... if installed?(id) # or return
+
   p = new(id)
   p.instance_eval(&block)
```

I think, it is difficult to find 2. by administrator if only return.

#3 - 2018-01-15 03:13 - Akipii Oga

+1 .
I agree.
I want all plugin and installation of Redmine main body to change to Gem. Also, I would like the installation work to be easy.

#4 - 2018-02-03 11:44 - Toshi MARUYAMA

Sho HASHIMOTO wrote:

> I think, it is difficult to find 2. by administrator if only return.

Can we raise exception if plugin name is duplicate?

#5 - 2018-03-24 04:44 - Sho HASHIMOTO

> > I think, it is difficult to find 2. by administrator if only return.
>
> Can we raise exception if plugin name is duplicate?

OK. I created another ticket. #28412

#6 - 2018-03-31 04:33 - Go MAEDA

- Category set to Plugin API

#7 - 2023-08-23 16:12 - Takashi Kato

- File 0001-Enable-to-use-gem-as-a-Redmine-plugin.patch added

- File 0002-Add-redmine_plugin_gem-generator.patch added

- File 0003-Sohw-gems-badge-on-plugin-list.patch added

- File 0004-Add-dependency-test.patch added

Patches for Redmine 5.0 or later (Rails 6.1 or later) are available here(tested in r22282).

Making plugins as gems allows you to install all plugins at once by listing them in the Gemfile.extension and running bundle install.
This can be especially useful in a Docker environment where you don't need to create a separate script to install plugins.
On the other hand, plugins placed under plugins/ directory will still work. I will refer to current plugins as **"filesystem plugin"** and gemified plugins as **"gem plugin"**.

If there's a **"filesystem plugin"** and a **"gem plugin"** with the same ID, the **"filesystem plugin"** takes priority
To install a **"gem plugin"**, add its name to Gemfile.extension. Note that the reason why Gemfile.extension is not named Gemfile.plugin is that we assume that themes will also be gemmed in the future.

For a **"gem plugin"**, the plugin information is divided into two files: init.rb and gemspec. Basic information like plugin name and author goes in gemspec, while Redmine-specific information like menus and settings stays in init.rb.

As a trial, I converted redmine-view-customize into a **"gem plugin"**. It works well. https://github.com/tohosaku/redmine-view-customize/tree/rubygems
The gem name, Redmine plugin ID, git repository name, and class names under the lib directory do not need to match. However, you can match all of

them.

gem name	redmine-view-cusomize
git repository name	redmine-view-cusomize
Redmine::Plugin#id	view_customize
class name	redmine_view_cusomize

On the other hand, if you try to register a plugin with a different gem name and the same plugin id, an error will occur.

Differences from the previous version

"filesystem plugin" allowed adding a gem to Redmine itself by adding the gem in the plugin's Gemfile. This is because Redmine was reading each plugin's Gemfile and requiring it as a dependent gem of redmine itself. gem plugin requires an add_runtime_dependency to be listed in each gemspec. The dependencies are automatically loaded when the plugin is registered with Redmine.

Creating a new plugin

```
bin/rails g redmine_plugin_gem foo
```

When creating a gem, it is common to use the bundle gem command to create a template, but since there is a difference between a normal gem and a **"gem plugin"** in the way the gemspec is described below, we recommend using bin/rails g to create the template.

Notes on gemspec for "gem plugin"

- In the gemspec file, Do not use variables, constants, etc. defined by gem (eg. Foo::VERSION). gemspec is evaluated before rails initialization is completed. redmine plugin variables and constants are supposed to be managed by the zeitwerk autoloader after rails is started, so using them at initialization time will cause errors.
- Redmine plugin dependencies can also be described in gemspec using "add_runtime_dependency".
- The spec.metadata["redmine_plugin_id"] = "plugin id" is required as metadata in gemspec.
- The following attributes that were previously listed in init.rb will be overwritten by the corresponding attributes in gemspec.

attributes of init.rb	**attributes of gemspec**
name	summary
author	authors
description	description
version	version
url	homepage
author_url	metadata['author_url']

UI Changes

Added the ability to distinguish **"gem plugin"** or not in the list of plugins.

How to develop “filesystem plugin” and “gem plugin” in one development branch

- Place the gem plugin directory in the plugins directory, and enter the gem information required by the **"filesystem plugin"** for Redmine 4.2 and Redmine 5.0 in the PluginGemfile. Gemfile is ignored. Dependent gem information must be included in both the PluginGemfile and gemspec.
- To create a plugin that works with both Redmine 4.2 and Redmine 5.0 or later, the init.rb must look like this

```
if Rails.version > '6.0' && Rails.autoloaders.zeitwerk_enabled?
    # for Redmine5.0 or later
else
  Rails.configuration.to_prepare do
    # for Redmine4.*
  end
end
```

#8 - 2024-01-02 19:37 - Takashi Kato

- File 0001v2-Enable-to-use-gem-as-a-Redmine-plugin.patch added

Updated patch 0001, can be applied to r22583

Files

support_gem_plugin.patch	1.22 KB	2017-12-02	Sho HASHIMOTO
0001-Enable-to-use-gem-as-a-Redmine-plugin.patch	18.9 KB	2023-08-23	Takashi Kato
0002-Add-redmine_plugin_gem-generator.patch	9.67 KB	2023-08-23	Takashi Kato
0003-Sohw-gems-badge-on-plugin-list.patch	1.6 KB	2023-08-23	Takashi Kato
0004-Add-dependency-test.patch	1.61 KB	2023-08-23	Takashi Kato
0001v2-Enable-to-use-gem-as-a-Redmine-plugin.patch	19.2 KB	2024-01-02	Takashi Kato