Ruby - Misc #15486

Default gems README.md

12/29/2018 12:12 PM - zverok (Victor Shepelev)

Status:	Feedback
Priority:	Normal
Assignee:	hsbt (Hiroshi SHIBATA)

Description

While working on [Ruby Changelog](#), I noticed the following.

A lot of parts of stdlib is extracted currently into "default gems". This, in my understanding, means (amongst other things) their development is now in separate repositories on GitHub, and their development is semi-independent.

The problem I'd like to emphasize is **their README is unclear** about "what is it". Let's look at ostruct for example: https://github.com/ruby/ostruct. There are two huge problems:

- Stumbling upon this repo, how should one know it is a) a part of Ruby stdlib? and b) the authoritative source of this part (and not a mirror of the code in ruby/ruby repo)?
- There is some basic documentation explaining the usage of the library, but it would NOT be rendered anywhere in the standard library docs, so it is basically useless (which is not obvious for repo contributors).

I believe that for standard library gems the README should look somehow as following:

> This is the development repository of Ruby ostruct (OpenStruct) standard library.
>
> The library provides an OpenStruct data structure, similar to a Hash, that allows the definition of arbitrary attributes with their accompanying values.
>
> Canonical library docs: [OpenStruct](#)
>
> Before participating in the development of ostruct, you should know the following:
>
> - The development process is standard "fork => commit => pull request"
> - New versions of the standard library are released with new versions of Ruby
> - Versioning policy: ...
> - Code quality policy: ...

The last two points should probably link to the common documentation for all "default gems"... Well, as well as the whole text. So, all in all, there should be README template, where only gem names, short descriptions, and links to "canonical docs" are different (and maybe some code structure/contribution details for bigger libraries).

WDYT?

Related issues:

Related to Ruby - Bug #15376: Default gems: how will it work exactly?	**Closed**
Related to Ruby - Bug #15500: Behavior of require method in 2.5 is different ...	**Closed**

History

#1 - 12/29/2018 06:26 PM - marcandre (Marc-Andre Lafortune)

I've stared improving the README for the [matrix gem](#).

Feel free to provide PRs for it, and it could serve as a model for the others.

#2 - 12/29/2018 06:28 PM - marcandre (Marc-Andre Lafortune)

- Related to Bug #15376: Default gems: how will it work exactly? added

#3 - 12/29/2018 11:20 PM - hsbt (Hiroshi SHIBATA)

- Status changed from Open to Assigned

- Assignee set to hsbt (Hiroshi SHIBATA)

#4 - 12/30/2018 10:26 AM - colby (Colby Swandale)

I just come across this issue. I wanted to point out that I've been working on the:

- https://github.com/ruby/logger
- https://github.com/ruby/fileutils
- https://github.com/ruby/irb
- https://github.com/ruby/rake

Repositories for the past few weeks and improving whatever i can. I find that the READMEs on the logger, fileutils, and irb projects can defiantly be improved. I've made PRs and raised issues in Github in each project.

Some ideas i have are having generated documentation be published in each project on Github. Have more comprehensive examples in the README.

I could defiantly use some help to get some ideas on how to best do this and also help to improve the rest of the gems.

Thanks!

#5 - 12/31/2018 02:38 PM - rbjl (Jan Lelis)

I'd like to help out, too.

Maybe we can have a second file README-DEFAULT-GEM.md which is linked to by the README.md. The README-DEFAULT-GEM would explain how default gems work etc. and would be same for all default gems and would be updated in an automated way (i.e. one script that updates all). The README would then only contain gem specific examples, explanations, etc.

#6 - 01/02/2019 10:37 PM - rbjl (Jan Lelis)

As an example, I've created such a "default gem explanation" file as PR to the matrix library, see here: https://github.com/ruby/matrix/pull/7/files

What do you all think about this approach?

#7 - 01/03/2019 06:20 AM - marcandre (Marc-Andre Lafortune)

- Related to Bug #15500: Behavior of require method in 2.5 is different from 2.4 and 2.6 added

#8 - 01/04/2019 06:57 AM - duerst (Martin Dürst)

rbjl (Jan Lelis) wrote:

> As an example, I've created such a "default gem explanation" file as PR to the matrix library, see here: https://github.com/ruby/matrix/pull/7/files
>
> What do you all think about this approach?

I think in general, it is a good idea to add such information.

But we have to be careful. The main issues I'm seeing with the actual PR are:

1. Where and how to submit bug reports, patches, feature requests,... may be different from gem to gem.

2. You mention https://stdgems.org/. This looks good, and seems to be up to date (I didn't do any actual checks, though). But what guarantee, if any, do we have that it will still be up to date in some years? It says "You can find out which version of **matrix** is included in your version of Ruby on stdgems.org/matrix.". Don't we have a way to check a gem version directly? If we don't, we should create one. I think it's okay to write something like "https://stdgems.org/ may provide additional information about gem versions", but adding yet another Web site that we depend on may be suboptimal.

#9 - 01/09/2019 02:38 PM - rbjl (Jan Lelis)

Thank you for the feedback. In the GH issue the general opinion is to have a single README file with all the info.

duerst (Martin Dürst) wrote:

> But we have to be careful. The main issues I'm seeing with the actual PR are:
>
> > 1. Where and how to submit bug reports, patches, feature requests,... may be different from gem to gem.

This is correct, although it might not be overly complex. If I understood hsbt correctly, the goal is that all future bug reports go to gem repository directly (i.e. a repo on GitHub). Until we get there we could add another sentence with a pointer to ruby-lang ala "You can also submit your bug reports and feature requests to bugs.ruby-lang.org"

Also, if we'd choose to generate README(part)s via templates, the correct point of maintenance could be integrated easily.

1. You mention https://stdgems.org/. This looks good, and seems to be up to date (I didn't do any actual checks, though). But what guarantee, if any, do we have that it will still be up to date in some years? It says "You can find out which version of **matrix** is included in your version of Ruby on stdgems.org/matrix.". Don't we have a way to check a gem version directly? If we don't, we should create one. I think it's okay to write something like "https://stdgems.org/ may provide additional information about gem versions", but adding yet another Web site that we depend on may be suboptimal.

You are right, there is no guarantee and I agree with your thoughts. Nevertheless, I've also put the raw data on GitHub, so it does not depend on me solely: https://github.com/janlelis/stdgems. If there is an interest in having one I would be open to build an API around this data, so it can be requested from inside Ruby.

#10 - 01/10/2019 06:10 AM - duerst (Martin Dürst)

Hello Jan,

You are right, there is no guarantee and I agree with your thoughts. Nevertheless, I've also put the raw data on GitHub, so it does not depend on me solely: https://github.com/janlelis/stdgems. If there is an interest in having one I would be open to build an API around this data, so it can be requested from inside Ruby.

Thanks for contributing to this issue. I think it's good to have this data on GitHub. It may also be a good idea to have this data accessible by a Web API.

But I still have a very fundamental question: Why isn't this information available by introspection, without having to go over the net to a Web API? Ruby itself should not have any problem providing this information, I'd guess. If I'm wrong, what's the problem?

#11 - 01/10/2019 08:11 AM - duerst (Martin Dürst)

duerst (Martin Dürst) wrote:

But I still have a very fundamental question: Why isn't this information available by introspection, without having to go over the net to a Web API? Ruby itself should not have any problem providing this information, I'd guess. If I'm wrong, what's the problem?

I found $LOADED_FEATURES, but it doesn't contain the necessary information::

```
require 'bigdecimal'
 p $LOADED_FEATURES.grep(/bigdecimal/)
```

=> ["/usr/local/lib/ruby/2.6.0/x86_64-cygwin/bigdecimal.so"]

#12 - 03/11/2019 12:19 PM - hsbt (Hiroshi SHIBATA)

- Status changed from Assigned to Feedback

We welcome a patch for README.md of default gems on GitHub directly like https://bugs.ruby-lang.org/issues/15486#note-4.