# How to contribute

Community involvement is essential to RubyGems. We want to keep it as easy as

possible to contribute changes. There are a few guidelines that we need

contributors to follow to reduce the time it takes to get changes merged in.

## Guidelines

1. New features should be coupled with tests.

2. Ensure that your code blends well with ours:

\* No trailing whitespace

\* Match indentation (two spaces)

\* Match coding style (run `rake rubocop`)

3. If any new files are added or existing files removed in a commit or PR,

please update the `Manifest.txt` accordingly. This can be done by running

`rake update\_manifest`

4. Don't modify the history file or version number.

5. If you have any questions, Feel free to join us on Slack, you can register

by signing up at http://slack.bundler.io or file an issue here:

http://github.com/rubygems/rubygems/issues

For more information and ideas on how to contribute to RubyGems ecosystem, see

here: https://guides.rubygems.org/contributing/

## Getting Started

$ rake setup

$ rake test

> Optional you can configure git hooks with: rake git\_hooks

To run commands like `gem install` from the repo:

$ ruby -Ilib bin/gem install

To run bundler test:

$ cd bundler

$ bin/rake spec:deps

$ bin/rspec spec

## Issues

RubyGems uses labels to track all issues and pull requests. In order to

provide guidance to the community this is documentation of how labels are used

in the rubygems repository.

### Contribution

These labels are made to guide contributors to issue/pull requests that they

can help with. That are marked with a light gray `contribution: \*`

\* \*\*small\*\* - The issue described here will take a small amount of work to

resolve, and is a good option for a new contributor

\* \*\*unclaimed\*\* - The issue has not been claimed for work, and is awaiting

willing volunteers!

### Type

Issues might have a light green `type: \*` label, which describes the type of

the issue.

\* \*\*bug report\*\* - An issue describing a bug in rubygems. This would be

something that is broken, confusing, unexpected behavior etc.

\* \*\*feature request\*\* - An issue describing a request for a new feature or

enhancement.

\* \*\*question\*\* - An issue that is a more of a question than a call for

specific changes in the codebase.

\* \*\*cleanup\*\* - An issue that proposes cleanups to the code base without

fixing a bug or implementing a feature.

\* \*\*major bump\*\* - This issue request requires a major version bump

\* \*\*administrative\*\* - This issue relates to administrative tasks that need

to take place as it relates to rubygems

\* \*\*documentation\*\* - This issue relates to improving the documentation for

in this repo. Note that much of the rubygems documentation is here:

https://github.com/rubygems/guides

Pull request might have a light orange `rubygems: \*` or a light blue `bundler:

\*` label which describes the pull request according to the following criteria:

\* \*\*security fix\*\* - A pull request that fixes a security issue.

\* \*\*breaking change\*\* - A pull request including any change that requires a

major version bump.

\* \*\*major enhancement\*\* - A pull request including a backwards compatible

change worth a special mention in the changelog

\* \*\*deprecation\*\* - A pull request that introduces a deprecation.

\* \*\*feature\*\* - A pull request implementing a feature request.

\* \*\*deprecation\*\* - A pull request that implements a performance improvement.

\* \*\*documentation\*\* - A pull request introducing documentation improvements

worth mentioning to end users.

\* \*\*minor enhancements\*\* - A pull request introducing small but user visible changes.

\* \*\*bug fix\*\* - A pull request that fixes a bug report.

In the case of `bundler`, these labels are set by maintainers on PRs and have

special importance because they are used to automatically build the changelog.

### Workflow / Status

The light yellow `status: \*` labels that indicate the state of an issue,

where it is in the process from being submitted to being closed. These are

listed in rough progression order from submitted to closed.

\* \*\*triage\*\* - This is an issue or pull request that needs to be properly

labeled by by a maintainer.

\* \*\*confirmed\*\* - This issue/pull request has been accepted as valid, but is

not yet immediately ready for work.

\* \*\*ready\*\* - An issue that is available for collaboration. This issue

should have existing discussion on the problem, and a description of how

to go about solving it.

\* \*\*working\*\* - An issue that has a specific invidual assigned to and

planning to do work on it.

\* \*\*user feedback required\*\* - The issue/pull request is blocked pending

more feedback from an end user

\* \*\*blocked / backlog\*\* - the issue/pull request is currently unable to move

forward because of some specific reason, generally this will be a reason

that is outside RubyGems or needs feedback from some specific individual

or group, and it may be a while before something it is resolved.

### Closed Reason

Reasons are why an issue / pull request was closed without being worked on or

accepted. There should also be more detailed information in the comments. The

closed reason labels are maroon `closed: \*`.

\* \*\*duplicate\*\* - This is a duplicate of an existing bug. The comments must

reference the existing issue.

\* \*\*abandonded\*\* - This is an issue/pull request that has aged off, is no

longer applicable or similar.

\* \*\*declined\*\* - An issue that won't be fixed/implemented or a pull request

that is not accepted.

\* \*\*deprecated\*\* - An issue/pull request that no longer applies to the

actively maintained codebase.

\* \*\*discussion\*\* - An issue/pull that is no longer about a concrete change,

and is instead being used for discussion.

### Categories

These are aspects of the codebase, or what general area the issue or pull

request pertains too. Not all issues will have a category. All categorized

issues have a blue `category: \*` label.

\* \*\*gemspec\*\* - related to the gem specification itself

\* \*\*API\*\* - related to the public supported rubygems API. This is the code

API, not a network related API.

\* \*\*command\*\* - related to something in `Gem::Commands`

\* \*\*install\*\* - related to gem installations

\* \*\*documentation\*\* - related to updating / fixing / clarifying

documentation or guides

### Platforms

If an issue or pull request pertains to only one platform, then it should have

an appropriate purple `platform: \*` label. Current platform labels:

\*\*windows\*\*, \*\*java\*\*, \*\*osx\*\*, \*\*linux\*\*

### Git

Please sign your commits. Although not required in order for you to contribute,

it ensures that any code submitted by you wasn't altered while you were

transferring it, and proves that it was you who submitted it and not someone

else.

Please see https://git-scm.com/book/en/v2/Git-Tools-Signing-Your-Work or

https://help.github.com/en/articles/signing-commits for details on how to

to generate a signature and automatically sign your commits.