# Contributing to Puma

By participating in this project, you agree to follow the [code of conduct].

[code of conduct]: https://github.com/puma/puma/blob/master/CODE\_OF\_CONDUCT.md

There are lots of ways to contribute to puma. Some examples include:

\* creating a [bug report] or [feature request]

\* verifying [existing bug reports] and adding [reproduction steps]

\* reviewing [pull requests] and testing the changes locally, on your own machine

\* writing or editing [documentation]

\* improving test coverage

\* fixing a [reproducing bug] or adding a new feature

[bug report]: https://github.com/puma/puma/issues/new?template=bug\_report.md

[feature request]: https://github.com/puma/puma/issues/new?template=feature\_request.md

[existing bug reports]: https://github.com/puma/puma/issues?q=is%3Aopen+is%3Aissue+label%3Aneeds-repro

[pull requests]: https://github.com/puma/puma/pulls

[documentation]: https://github.com/puma/puma/tree/master/docs

[reproduction steps]: https://github.com/puma/puma/blob/CONTRIBUTING.md#reproduction-steps

[reproducing bug]: https://github.com/puma/puma/issues?utf8=%E2%9C%93&q=is%3Aopen+is%3Aissue+label%3Abug

Newbies welcome! We would be happy to help you make your first contribution to a F/OSS project.

## Setup

Clone down the Puma repository.

You will need to install [ragel] (use Ragel version 7.0.0.9) to generate puma's extension code.

macOS:

```sh

brew install ragel

```

Linux:

```sh

apt-get install ragel

```

Install Ruby dependencies with:

```sh

bundle install

```

[ragel]: https://www.colm.net/open-source/ragel/

To run Puma, you will need to compile the native extension. To do this:

```sh

bundle exec rake compile

```

Then, you will be able to run Puma using your local copy with:

```sh

bundle exec bin/puma test/rackup/hello.ru

```

## Running tests

You can run the full test suite with:

```sh

bundle exec rake test:all

```

To run a single test file:

```sh

bundle exec ruby test/test\_binder.rb

```

Or use [`m`](https://github.com/qrush/m):

```sh

bundle exec m test/test\_binder.rb

```

... which can also be used to run a single test case:

```sh

bundle exec m test/test\_binder.rb:37

```

## How to contribute

Puma needs help in several areas.

\*\*The `contrib-wanted` label is applied to issues that maintainers think would be easier for first-time contributors.\*\*

\*\*Reproducing bug reports\*\*: The `needs-repro` label is applied to issues that have a bug report but no reproduction steps. You can help by trying to reproduce the issue and then posting how you did it.

\*\*Helping with our native extensions\*\*: If you can write C or Java, we could really use your help. Check out the issue labels for c-extensions and JRuby.

\*\*Fixing bugs\*\*: Issues with the `bug` label have working reproduction steps, which you can use to write a test and create a patch.

\*\*Writing features\*\*: Issues with the `feature` label are requests for new functionality. Write tests and code up our new feature!

\*\*Code review\*\*: Take a look at open pull requests and offer your feedback. Code review is not just for maintainers - we need your help and eyeballs!

\*\*Write documentation\*\*: Puma needs more docs in many areas, especially those where we have open issues labeled `docs`.

## Reproduction steps

Reproducing a bug helps identify the root cause of that bug so it can be fixed.

To get started, create a rackup file and config file and then run your test app

with:

```sh

bundle exec puma -C <path/to/config.rb> <path/to/rackup.ru>

```

As an example, using one of the test rack apps:

[`test/rackup/hello.ru`][rackup file], and one of the test config files:

[`test/config/settings.rb`][config], you would run the test app with:

```sh

bundle exec puma -C test/config/settings.rb test/rackup/hello.ru

```

There is also a Dockerfile available for reproducing Linux-specific issues. To use:

```sh

docker build -f tools/docker/Dockerfile -t puma .

docker run -p 9292:9292 -it puma

```

[rackup]: https://github.com/puma/puma/blob/master/test/rackup/hello.ru

[config]: https://github.com/puma/puma/blob/master/test/config/settings.rb

## Pull requests

Code contributions should generally include test coverage. If you aren't sure how to

test your changes, please open a pull request and leave a comment asking for

help.

If you open a pull request with a change that doesn't need to be noted in the

changelog ([`History.md`](History.md)), add the text `[changelog skip]` to the

pull request title to skip [the changelog

check](https://github.com/puma/puma/pull/1991).

## Bibliography/Reading

Puma can be a bit intimidating for your first contribution because there's a lot of concepts here that you've probably never had to think about before - Rack, sockets, forking, threads etc. Here are some helpful links for learning more about things related to Puma:

\* [Puma's Architecture docs](https://github.com/puma/puma/blob/master/docs/architecture.md)

\* [The Rack specification](https://www.rubydoc.info/github/rack/rack/file/SPEC)

\* The Ruby docs for IO.pipe, TCPServer/Socket.

\* [nio4r documentation](https://github.com/socketry/nio4r/wiki/Getting-Started)