



Sinatra

Fork me on GitHub

[README](#) [DOCUMENTATION](#) [BLOG](#) [CONTRIBUTE](#) [CREW](#) [CODE](#)
[ABOUT](#) [Gittip](#)

Contribute

Want to show Sinatra some love? Help out by contributing!

Find a bug?

Log it in our [issue tracker](#) or send a note to the [mailing list](#). Be sure to include all relevant information, like the versions of Sinatra and Ruby you're using. A [gist](#) of the code that caused the issue as well as any error messages are also very helpful.

Need help?

The [Sinatra mailing list](#) has over 900 subscribers, many of which are happy to help out newbies or talk about potential feature additions. You can also drop by the [#sinatra](#) channel on [irc.freenode.net](#).

Have a patch?

Bugs and feature requests that include patches are much more likely to get attention. Here are some guidelines that will help ensure your patch can be applied as quickly as possible:

1. **Use [Git](#) and [GitHub](#):** The easiest way to get setup is to fork the [sinatra/sinatra repo](#). Or, the [sinatra.github.com repo](#), if the patch is doc related.
2. **Write unit tests:** If you add or modify functionality, it must include unit tests. If you don't write tests, we have to, and this can hold up acceptance of the patch.
3. **Mind the [README](#):** If the patch adds or modifies a major feature, modify the [README.md](#) file to reflect that. Again, if you don't update the [README](#), we have to, and this holds up acceptance.
4. **Push it:** Once you're ready, push your changes to a topic branch and add a note to the ticket with the URL to your branch. Or, say something like, "you can find the patch on johndoe/foobranh". We also gladly accept GitHub [pull requests](#).

NOTE: *we will take whatever we can get.* If you prefer to attach diffs in emails to the mailing list, that's fine; but do know that *someone* will need to take the diff through the process described above and this can hold things up considerably.

Want to write docs?

The process for contributing to Sinatra's website, documentation or the book is the same as contributing code. We use git for versions control and GitHub to track patch requests.

- [The `sinatra.github.com` repo](#) is where the website sources are managed. There are almost always people in `#sinatra` that are happy to discuss, apply, and publish website patches.
- [The Book](#) has its own [git repository](#) and build process but is managed the same as the website and project codebase.
- [Sinatra Recipes](#) is a community project where anyone is free to contribute ideas, recipes and tutorials. Which also has its own [git repository](#).
- [The Introduction](#) is generated from Sinatra's [README file](#).
- If you want to help translating the documentation, there already is a [Japanese](#) and a [German](#) version of the README, which tend to fall behind the English version. Translations into other languages would also be appreciated.

Looking for something to do?

If you'd like to help out but aren't sure how, pick something that looks interesting from the [issues](#) list and hack on. Make sure to leave a comment on the ticket noting that you're investigating (a simple "Taking..." is fine).