# Thanks!

RailsBridge wouldn't be what it is without your help. Thanks for contributing.

We ask that contributions be made as pull requests via GitHub. If those words

are totally foreign to you,

[see here](#its-my-first-time-on-github-ever-what-do-i-do).

# When Submitting a Pull Request

\*Here's a couple of tricks to grease the wheels and make it easy for the

maintainers to love you. :heart:\*

## Before You Start!

- If you have an existing fork, please make sure it's up to date.

It just makes your life easier! If not, make sure you fork \*before\* cloning,

otherwise you'll need to spend some time juggling remotes.

Look at the section "Keep your fork synced" in GitHub's

[Fork A Repo](https://help.github.com/articles/fork-a-repo) article.

- Create a local topic branch before you start working. This branch is going to

be named for what you plan to change. `fix-typo-in-slides`, `move-resources`,

and `mountain-lion-support` are all good names for topic branches. If you've

never created a local branch before, you can use `git checkout -b

new-branch-name`.

## Before Submitting

- Please, please, \*please\* run `bundle exec rake` from the terminal before you submit. It

not only runs our test suite but also checks the docs for syntax errors.

- Push to a branch on GitHub. Just like you developed in a local branch, you

should push to a branch of your repo on GitHub as well. The `master` branch is

best used as a clean copy of the upstream docs repo in case you need to make

some unrelated changes. To push to a branch,

if your branch is named "fix-typo-in-slides",

use `git push origin fix-typo-in-slides`.

## Submitting a Pull Request

- Read the GitHub Guide on [Forking](https://guides.github.com/activities/forking/), especially the part about

[Pull Requests](https://guides.github.com/activities/forking/#making-a-pull-request).

- Remember, pull requests are submitted \*from\* your repo, but show up on the

\*upstream\* repo.

## Discussion and Waiting On a Merge

- Every pull request will receive a response from the team.

- Not every pull request will be merged as is.

- Not every pull request will be merged at all.

- If a pull request falls significantly behind master, we may ask that you close

it, rebase your changes off of master, and submit a new pull request.

- Feel free to "ping" the team by adding a short comment to your pull request

if it's been more than a week with no reply

## After your merge has been accepted

- go back to your fork and keep it up to date, e.g.

git checkout master

git pull upstream master

git push origin master

- you can also delete your topic branch if you like

git branch -dr fix-typo-in-slides

# It's My First Time on GitHub Ever What Do I Do?

Relax, you came to the right place. In order to contribute you'll need to be

able to familiarize yourself with some concepts from git and GitHub. It's going

to be a lot of information, but you're :sparkles:awesome:sparkles:! So you'll

be fine.

First, you'll need a GitHub account, which is totally free. You can sign up

[here](https://github.com/join).

Next, browse the [GitHub Help site](https://help.github.com) and the

[GitHub Guides](https://guides.github.com/). The Help Site is more technical, and the

Guides are very easy to understand tutorials.

You'll want to read about

[forking](https://help.github.com/articles/fork-a-repo) and then make your own

fork of [railsbridge/docs](https://github.com/railsbridge/docs). Once you've

done so, you can clone it and get started by reading up on [what to do when

submitting a pull request](#when-submitting-a-pull-request), and read up on

[pull requests](https://help.github.com/articles/using-pull-requests)

themselves.

If this is all too much, or you'd like a helping hand,

[@nuclearsandwich](https://github.com/nuclearsandwich) has volunteered to help

anyone who wants to contribute do so. His email is behind that profile link.

# Closing

If you haven't taken the time yet, go through the [Git Immersion lab](http://gitimmersion.com)

at <http://gitimmersion.com>. Do it. It's worth it no matter how much git-fu you have.

Also, [Pro Git](http://git-scm.com/book) is a great (and free!) book about Git.

We apologize for how long this document is! Hopefully it addressed

most of your concerns about git, contributing, and GitHub. Feel free

to ask more questions on the

[railsbridge-workshops](http://groups.google.com/group/railsbridge-workshops)

mailing list. And we're open to any suggestions about improvements,

including to this document.