## Contributing

First off, thank you for considering contributing to Active Admin. It's people

like you that make Active Admin such a great tool.

### Where do I go from here?

If you've noticed a bug or have a feature request, [make one][new issue]! It's

generally best if you get confirmation of your bug or approval for your feature

request this way before starting to code.

If you have a general question about activeadmin, you can post it on [Stack

Overflow], the issue tracker is only for bugs and feature requests.

### Fork & create a branch

If this is something you think you can fix, then [fork Active Admin] and create

a branch with a descriptive name.

A good branch name would be (where issue #325 is the ticket you're working on):

```sh

git checkout -b 325-add-japanese-translations

```

### Get the test suite running

Make sure you're using a recent ruby and have the `bundler` gem installed, at

least version `1.14.3`.

You'll also need chrome installed in order to run cucumber scenarios.

Now install the development dependencies:

```sh

bundle install

```

Then install javascript dependencies with [Yarn] (requires a current version of [Node.js]):

```sh

bin/yarn install

```

JS assets are located in `app/javascript/active\_admin`. The config will take care of compiling a complete bundle with [Rollup] using the `build` script and exported to `app/assets/javascripts/active\_admin/base.js` ready to be used by Sprockets.

To update javascript bundle run (add `-w` flag for watch mode):

```sh

bin/yarn build

```

Now you should be able to run the entire suite using:

```sh

bin/rake

```

The test run will generate a sample Rails application in `tmp/test\_apps` to run the

tests against.

If you want to test against a Rails version different from the latest, make sure

you use the correct Gemfile, for example:

```sh

export BUNDLE\_GEMFILE=gemfiles/rails\_60/Gemfile

```

\*\*Warning\*\* SCSS assets are aimed to be used indifferently with Sprockets \*\*and\*\* webpacker.

As such, make sure not to use any sass-rails directives such as `asset-url` or `image-url`.

### Implement your fix or feature

At this point, you're ready to make your changes! Feel free to ask for help;

everyone is a beginner at first :smile\_cat:

### View your changes in a Rails application

Active Admin is meant to be used by humans, not cucumbers. So make sure to take

a look at your changes in a browser.

To boot up a test Rails app:

```sh

bin/rake local server

```

This will automatically create a Rails app if none already exists, and store it

in the `tmp/development\_apps` folder.

You should now be able to open <http://localhost:3000/admin> in your browser.

You can log in using:

\*User\*: admin@example.com

\*Password\*: password

If you need to perform any other commands on the test application, just pass

them to the `local` rake task. For example, to boot the rails console:

```sh

bin/rake local console

```

Or to migrate the database, if you create a new migration or just play around

with the db:

```sh

bin/rake local db:migrate

```

### Get the style right

Your patch should follow the same conventions & pass the same code quality

checks as the rest of the project. `bin/rake lint` will give you feedback in

this regard. You can check & fix style issues by running each linter

individually. Run `bin/rake -T lint` to see the available linters.

### Make a Pull Request

At this point, you should switch back to your master branch and make sure it's

up to date with Active Admin's master branch:

```sh

git remote add upstream git@github.com:activeadmin/activeadmin.git

git checkout master

git pull upstream master

```

Then update your feature branch from your local copy of master, and push it!

```sh

git checkout 325-add-japanese-translations

git rebase master

git push --set-upstream origin 325-add-japanese-translations

```

Finally, go to GitHub and [make a Pull Request][] :D

CircleCI will run our test suite against all supported Rails versions. We care

about quality, so your PR won't be merged until all tests pass. It's unlikely,

but it's possible that your changes pass tests in one Rails version but fail in

another. In that case, you'll have to setup your development environment (as

explained in step 3) to use the problematic Rails version, and investigate

what's going on!

### Keeping your Pull Request updated

If a maintainer asks you to "rebase" your PR, they're saying that a lot of code

has changed, and that you need to update your branch so it's easier to merge.

To learn more about rebasing in Git, there are a lot of [good][git rebasing]

[resources][interactive rebase] but here's the suggested workflow:

```sh

git checkout 325-add-japanese-translations

git pull --rebase upstream master

git push --force-with-lease 325-add-japanese-translations

```

### Merging a PR (maintainers only)

A PR can only be merged into master by a maintainer if:

\* It is passing CI.

\* It has been approved by at least two maintainers. If it was a maintainer who

opened the PR, only one extra approval is needed.

\* It has no requested changes.

\* It is up to date with current master.

Any maintainer is allowed to merge a PR if all of these conditions are

met.

### Shipping a release (maintainers only)

Maintainers need to do the following to push out a release:

\* Switch to the master branch and make sure it's up to date.

\* Make sure you have [chandler] properly configured. Chandler is used to

automatically submit github release notes from the changelog right after

pushing the gem to rubygems.

\* Run one of `bin/rake release:prepare\_{prerelease,prepatch,patch,preminor,minor,premajor,major}`, push the result and create a PR.

\* Review and merge the PR. The generated changelog in the PR should include all user visible changes you intend to ship.

\* Run `bin/rake release` from the target branch once the PR is merged.

[chandler]: https://github.com/mattbrictson/chandler#2-configure-credentials

[Stack Overflow]: http://stackoverflow.com/questions/tagged/activeadmin

[new issue]: https://github.com/activeadmin/activeadmin/issues/new

[fork Active Admin]: https://help.github.com/articles/fork-a-repo

[make a pull request]: https://help.github.com/articles/creating-a-pull-request

[git rebasing]: http://git-scm.com/book/en/Git-Branching-Rebasing

[interactive rebase]: https://help.github.com/en/github/using-git/about-git-rebase

[shortcut reference links]: https://github.github.com/gfm/#shortcut-reference-link

[Rollup]: https://rollupjs.org/guide/en/#quick-start

[Yarn]: https://yarnpkg.com/en/docs/install

[Node.js]: https://nodejs.org/en/