# Contributing to Actionhero

First, \*\*THANK YOU\*\*.

Actionhero would not be the success it is today without the contributions of [many people](https://github.com/actionhero/actionhero/graphs/contributors). Actionhero is a community-led open source project. Thank you for taking the time to help out this open source project, and create something we can all use!

## The Actionhero community

Before you begin your contribution, please let us know in the Actionhero Slack team, which is available at https://slack.actionherojs.com. There are community members who can help you, and you may want to team up with another community member. This also helps ensure that more than one person isn't working on the same thing.

By engaging the with Actionhero community or contributing to this or related projects, you agree to our [Code of Conduct](https://github.com/actionhero/actionhero/blob/master/CODE\_OF\_CONDUCT.md).

## Pull Requests

All changes to Actionhero should be sent in as [Pull Requests](https://help.github.com/articles/about-pull-requests) to our [Github Project](https://github.com/actionhero/actionhero). Changes by any other method will be instantly rejected. GitHub allows us to coordinate and communicate in a single place. Pull requests also allow us to run our test suite against all new code to ensure that things still work the way they are supposed to after your change.

## Testing

Actionhero is a large project with lost of different servers and tools. We don't expect you to know about everything, that's why we have a robust test suite. This allows us to ensure that no matter who makes a change, Actionhero will continue to work the way it is supposed to.

With that in mind, all new features to Actionhero must also include additions to the test suite to ensure that in the future, we can maintain your work. When writing tests, write the smallest test that ensures that your work is tested. IE: if you write a new initializer, you probably can test the method directly and you don't need an action test or an integration test.

\*\*Every contribution to the codebase should have an associated test\*\*

Be sure that your changes pass the test suite! Run `npm test` to run the full test suite.

You will need redis and node.js installed. We also have an integration which relies on chromedriver (an automated chrome browser). You can install this on OSX via `brew install chromedriver` No other external dependencies are needed.

If you need help writing tests, please ask for help in the [slack team](http://slack.actionherojs.com)

## Linting

We use [prettier.js](https://prettier.io/) to manage our lint rules. We run `prettier` as part of our test suite, and your contributions must pass. Prettier is \_very\_ opinionated and inflexible such that we cannot inject our own opinions. There are no eslint/jshint files to manage in this project.

## Documentation

If your contribution adds a new feature of modifies an existing behavior, document your changes using [Typedoc](https://typedoc.org/). We use Typedoc to automatically document Actionhero, and build [https://docs.actionherojs.com](docs.actionherojs.com) on every push to the master branch or merge of your Pull Request. There are many plugins to help you with this, and are built into editors like VSCode.

If you are documenting code, inline JSdocs are preferred. The only exceptions are tutorials, which are stand-alone markdown files in the `./tutorials` directory of the [www.actionherojs.com project](https://github.com/actionhero/www.actionherojs.com). An example of a newly documented method would be:

```js

/\*\*

Sleep with a Promise.

`time` is the number of ms to sleep.

\*/

api.utils.sleep = (time: number) => {

return new Promise((resolve) => {

setTimeout(resolve, time);

});

};

```

The "marketing" www.actionherojs.com site [is built via this project separately](https://github.com/actionhero/www.actionherojs.com).