# How to contribute to Phinx

Phinx relies heavily on external contributions in order to make it the best database migration

tool possible. Without the support of our 130+ contributors, we wouldn't be where we are today!

We encourage anyone to submit documentation enhancements and code.

Issues, feature requests, and bugs should be submitted using the Github issue tool:

https://github.com/cakephp/phinx/issues.

This document briefly outlines the requirements to contribute code to Phinx.

## Considerations

Before you submit your pull request take a moment to answer the following questions.

Answering '\*\*YES\*\*' to all questions will increase the likelihood of your PR being accepted!

\* Have I implemented my feature for as many database adapters as possible?

\* Does my new feature improve Phinx's performance or keep it consistent?

\* Does my feature fit within the database migration space?

\* Is the code entirely my own and free from any commercial licensing?

\* Am I happy to release my code under the MIT license?

\* Is my code formatted using the [PSR-2](https://github.com/php-fig/fig-standards/blob/master/accepted/PSR-2-coding-style-guide.md) coding standard?

\*\*Note:\*\* We accept bug fixes much faster into our development branch than new features.

## Getting Started

Great, so you want to contribute. Let's get started:

1. Start by forking Phinx on GitHub: https://github.com/cakephp/phinx

1. Clone your repository to a local directory on your development box.

1. If you do not have [Composer](https://getcomposer.org) set up already, install it:

```

curl -sS https://getcomposer.org/installer | php

```

1. Change to your Phinx clone directory and pull the necessary dependencies:

```

php composer.phar install

```

1. Copy the `phpunit.xml.dist` template to `phpunit.xml`:

```

cp phpunit.xml.dist phpunit.xml

```

And change the configuration to suit your environment. If you are not using any particular adapter you can disable it in the `phpunit.xml` file.

1. Run the unit tests locally to ensure they pass:

```

php vendor/bin/phpunit --config phpunit.xml

```

1. Write the code and unit tests for your bug fix or feature.

1. Add any relevant documentation.

1. Run the unit tests again and ensure they pass.

1. Open a pull request on the Github project page. Ensure the code is being merged into `master`.

## Getting Started with docker

If you do not wish to install MySQL and Postgres locally to run unit tests

install [docker-compose](https://docs.docker.com/compose/) for your platform.

1. Download all images, build and start development container:

```

docker-compose run --rm phinx

```

1. Install dependencies:

```

composer update

```

1. Run unittest:

```

vendor/bin/phpunit

```

## Documentation

The Phinx documentation is stored in the \*\*docs\*\* directory using the [RestructedText](http://docutils.sourceforge.net/rst.html)

format. All documentation merged to `master` is automatically published to the Phinx documentation site available at http://docs.phinx.org.