# Contributing

We'd love you to help out with mockery and no contribution is too small.

## Reporting Bugs

Issues can be reported on the [issue

tracker](https://github.com/padraic/mockery/issues). Please try and report any

bugs with a minimal reproducible example, it will make things easier for other

contributors and your problems will hopefully be resolved quickly.

## Requesting Features

We're always interested to hear about your ideas and you can request features by

creating a ticket in the [issue

tracker](https://github.com/padraic/mockery/issues). We can't always guarantee

someone will jump on it straight away, but putting it out there to see if anyone

else is interested is a good idea.

Likewise, if a feature you would like is already listed in

the issue tracker, add a :+1: so that other contributors know it's a feature

that would help others.

## Contributing code and documentation

We loosely follow the

[PSR-1](https://github.com/php-fig/fig-standards/blob/master/accepted/PSR-1-basic-coding-standard.md)

and

[PSR-2](https://github.com/php-fig/fig-standards/blob/master/accepted/PSR-2-coding-style-guide.md) coding standards,

but we'll probably merge any code that looks close enough.

\* Fork the [repository](https://github.com/padraic/mockery) on GitHub

\* Add the code for your feature or bug

\* Add some tests for your feature or bug

\* Optionally, but preferably, write some documentation

\* Optionally, update the CHANGELOG.md file with your feature or

[BC](http://en.wikipedia.org/wiki/Backward\_compatibility) break

\* Send a [Pull

Request](https://help.github.com/articles/creating-a-pull-request) to the

correct target branch (see below)

If you have a big change or would like to discuss something, create an issue in

the [issue tracker](https://github.com/padraic/mockery/issues) or jump in to

\#mockery on freenode

Any code you contribute must be licensed under the [BSD 3-Clause

License](http://opensource.org/licenses/BSD-3-Clause).

## Target Branch

Mockery may have several active branches at any one time and roughly follows a

[Git Branching Model](https://igor.io/2013/10/21/git-branching-model.html).

Generally, if you're developing a new feature, you want to be targeting the

master branch, if it's a bug fix, you want to be targeting a release branch,

e.g. 0.8.

## Testing Mockery

To run the unit tests for Mockery, clone the git repository, download Composer using

the instructions at [http://getcomposer.org/download/](http://getcomposer.org/download/),

then install the dependencies with `php /path/to/composer.phar install`.

This will install the required PHPUnit and Hamcrest dev dependencies and create the

autoload files required by the unit tests. You may run the `vendor/bin/phpunit` command

to run the unit tests. If everything goes to plan, there will be no failed tests!

## Debugging Mockery

Mockery and its code generation can be difficult to debug. A good start is to

use the `RequireLoader`, which will dump the code generated by mockery to a file

before requiring it, rather than using eval. This will help with stack traces,

and you will be able to open the mock class in your editor.

``` php

// tests/bootstrap.php

Mockery::setLoader(new Mockery\Loader\RequireLoader(sys\_get\_temp\_dir()));

```