# Contribution guidelines #

## Reporting issues ##

Make sure that what you are experiencing is actually an error and that it lies

with homeshick (often it can be a git configuration error)

### Questions ###

If you have a question be sure to read

[the documentation](https://github.com/andsens/homeshick/wiki) first.

Often you will find the answer to it there.

### Description ###

As with bug reports everywhere else:

\* state the action(s) you took

\* explain what outcome you expected

\* describe the actual result

You will also need to report which operating system you encountered the issue on

and which shell you used (type `echo $SHELL` in your terminal if you are unsure).

### Reproducing ###

Unless you ran in to a [heisenbug](http://en.wikipedia.org/wiki/Heisenbug),

it should be possible to reproduce the

bug in a testing environment. To that end run

`$HOME/.homesick/repos/homeshick/test/interactive` and reproduce the bug. This

script drops you into a new shell where `$HOME` is set to an (almost) empty

temporary folder. If you cannot reproduce the bug there, the error is likely

with your setup and not homeshick. Otherwise attach the commands you executed

in that environment to the issue.

## Pull requests ##

### Branching

\*\*Work from and create pull requests on `development`, not `master`.\*\*

`master` always represents the latest release since that is the way homeshick

updates itself. The `development` branch is where work is done for the next

release version of homeshick.

### Code style ###

\* Indent with tabs and align with spaces.

\* Always use double brackets for `if` blocks

\* Run your changes through

[shellcheck](https://www.shellcheck.net/) with `test/shellcheck`

and the [bats](https://github.com/sstephenson/bats) testsuite with `test/run`

Use the supplied [editorconfig](http://editorconfig.org) file.

Most editors have [editorconfig plugins](http://editorconfig.org/#download)

to apply these settings.

### Content ###

\*\*Every PR should only contain one feature change, bug fix or typo correction.\*\*

Commits should be atomic units of work, if they are not you should rebase them

so that they are (typo corrections from a previous change for example do not

justify a commit).

### Description ###

The PR should clearly describe what problem the change fixes.

A feature addition with no justification and use-case will be rejected.

### Testing ###

Unless the code-change is a refactor, you should always add unit tests. When

fixing a bug there should be a new test case that fails with the old code and

succeeds with the new code. When introducing a new feature, it should be

tested extensively, a single test case will not suffice.

Note that bats does not fail a test case when using double brackets.

To assert variable values and file existance you \*must\* use single brackets!

Also consider negative test cases (e.g. what happens when a non-existing

castlename is passed as an argument?).

You can read about the details of the testing framework in the

[testing documentation](https://github.com/andsens/homeshick/wiki/Testing).