# CONTRIBUTING

## RESOURCES

If you wish to contribute to this project, please be sure to

read/subscribe to the following resources:

- [Coding Standards](https://github.com/zendframework/zend-coding-standard)

- [Forums](https://discourse.zendframework.com/c/contributors)

- [Chat](https://zendframework-slack.herokuapp.com)

- [Code of Conduct](CODE\_OF\_CONDUCT.md)

If you are working on new features or refactoring

[create a proposal](https://github.com/zendframework/zend-code/issues/new).

## RUNNING TESTS

To run tests:

- Clone the repository:

```console

$ git clone git://github.com/zendframework/zend-code.git

$ cd zend-code

```

- Install dependencies via composer:

```console

$ composer install

```

If you don't have `composer` installed, please download it from https://getcomposer.org/download/

- Run the tests using the "test" command shipped in the `composer.json`:

```console

$ composer test

```

You can turn on conditional tests with the `phpunit.xml` file.

To do so:

- Copy `phpunit.xml.dist` file to `phpunit.xml`

- Edit `phpunit.xml` to enable any specific functionality you

want to test, as well as to provide test values to utilize.

## Running Coding Standards Checks

First, ensure you've installed dependencies via composer, per the previous

section on running tests.

To run CS checks only:

```console

$ composer cs-check

```

To attempt to automatically fix common CS issues:

```console

$ composer cs-fix

```

If the above fixes any CS issues, please re-run the tests to ensure

they pass, and make sure you add and commit the changes after verification.

## Recommended Workflow for Contributions

Your first step is to establish a public repository from which we can

pull your work into the master repository. We recommend using

[GitHub](https://github.com), as that is where the component is already hosted.

1. Setup a [GitHub account](https://github.com/), if you haven't yet

2. Fork the repository (https://github.com/zendframework/zend-code)

3. Clone the canonical repository locally and enter it.

```console

$ git clone git://github.com/zendframework/zend-code.git

$ cd zend-code

```

4. Add a remote to your fork; substitute your GitHub username in the command

below.

```console

$ git remote add {username} git@github.com:{username}/zend-code.git

$ git fetch {username}

```

### Keeping Up-to-Date

Periodically, you should update your fork or personal repository to

match the canonical ZF repository. Assuming you have setup your local repository

per the instructions above, you can do the following:

```console

$ git checkout master

$ git fetch origin

$ git rebase origin/master

# OPTIONALLY, to keep your remote up-to-date -

$ git push {username} master:master

```

If you're tracking other branches -- for example, the "develop" branch, where

new feature development occurs -- you'll want to do the same operations for that

branch; simply substitute "develop" for "master".

### Working on a patch

We recommend you do each new feature or bugfix in a new branch. This simplifies

the task of code review as well as the task of merging your changes into the

canonical repository.

A typical workflow will then consist of the following:

1. Create a new local branch based off either your master or develop branch.

2. Switch to your new local branch. (This step can be combined with the

previous step with the use of `git checkout -b`.)

3. Do some work, commit, repeat as necessary.

4. Push the local branch to your remote repository.

5. Send a pull request.

The mechanics of this process are actually quite trivial. Below, we will

create a branch for fixing an issue in the tracker.

```console

$ git checkout -b hotfix/9295

Switched to a new branch 'hotfix/9295'

```

... do some work ...

```console

$ git commit

```

... write your log message ...

```console

$ git push {username} hotfix/9295:hotfix/9295

Counting objects: 38, done.

Delta compression using up to 2 threads.

Compression objects: 100% (18/18), done.

Writing objects: 100% (20/20), 8.19KiB, done.

Total 20 (delta 12), reused 0 (delta 0)

To ssh://git@github.com/{username}/zend-code.git

b5583aa..4f51698 HEAD -> master

```

To send a pull request, you have two options.

If using GitHub, you can do the pull request from there. Navigate to

your repository, select the branch you just created, and then select the

"Pull Request" button in the upper right. Select the user/organization

"zendframework" (or whatever the upstream organization is) as the recipient.

#### What branch to issue the pull request against?

Which branch should you issue a pull request against?

- For fixes against the stable release, issue the pull request against the

"master" branch.

- For new features, or fixes that introduce new elements to the public API (such

as new public methods or properties), issue the pull request against the

"develop" branch.

### Branch Cleanup

As you might imagine, if you are a frequent contributor, you'll start to

get a ton of branches both locally and on your remote.

Once you know that your changes have been accepted to the master

repository, we suggest doing some cleanup of these branches.

- Local branch cleanup

```console

$ git branch -d <branchname>

```

- Remote branch removal

```console

$ git push {username} :<branchname>

```