# Contributing to CodeIgniter

CodeIgniter is a community driven project and accepts contributions of code and documentation from the community. These contributions are made in the form of Issues or [Pull Requests](http://help.github.com/send-pull-requests/) on the [CodeIgniter repository](https://github.com/bcit-ci/CodeIgniter) on GitHub.

Issues are a quick way to point out a bug. If you find a bug or documentation error in CodeIgniter then please check a few things first:

1. There is not already an open Issue

2. The issue has already been fixed (check the develop branch, or look for closed Issues)

3. Is it something really obvious that you can fix yourself?

Reporting issues is helpful but an even better approach is to send a Pull Request, which is done by "Forking" the main repository and committing to your own copy. This will require you to use the version control system called Git.

## Guidelines

Before we look into how, here are the guidelines. If your Pull Requests fail

to pass these guidelines it will be declined and you will need to re-submit

when you’ve made the changes. This might sound a bit tough, but it is required

for us to maintain quality of the code-base.

### PHP Style

All code must meet the [Style Guide](https://codeigniter.com/userguide3/general/styleguide.html), which is

essentially the [Allman indent style](https://en.wikipedia.org/wiki/Indent\_style#Allman\_style), underscores and readable operators. This makes certain that all code is the same format as the existing code and means it will be as readable as possible.

### Documentation

If you change anything that requires a change to documentation then you will need to add it. New classes, methods, parameters, changing default values, etc are all things that will require a change to documentation. The change-log must also be updated for every change. Also PHPDoc blocks must be maintained.

### Compatibility

CodeIgniter recommends PHP 5.5 or newer to be used, but it should be

compatible with PHP 5.2.4 so all code supplied must stick to this

requirement. If PHP 5.3 (and above) functions or features are used then

there must be a fallback for PHP 5.2.4.

### Branching

CodeIgniter uses the [Git-Flow](https://nvie.com/posts/a-successful-git-branching-model/) branching model which requires all pull requests to be sent to the "develop" branch. This is

where the next planned version will be developed. The "master" branch will always contain the latest stable version and is kept clean so a "hotfix" (e.g: an emergency security patch) can be applied to master to create a new version, without worrying about other features holding it up. For this reason all commits need to be made to "develop" and any sent to "master" will be closed automatically. If you have multiple changes to submit, please place all changes into their own branch on your fork.

One thing at a time: A pull request should only contain one change. That does not mean only one commit, but one change - however many commits it took. The reason for this is that if you change X and Y but send a pull request for both at the same time, we might really want X but disagree with Y, meaning we cannot merge the request. Using the Git-Flow branching model you can create new branches for both of these features and send two requests.

### Signing

You must sign your work, certifying that you either wrote the work or otherwise have the right to pass it on to an open source project. git makes this trivial as you merely have to use `--signoff` on your commits to your CodeIgniter fork.

`git commit --signoff`

or simply

`git commit -s`

This will sign your commits with the information setup in your git config, e.g.

`Signed-off-by: John Q Public <john.public@example.com>`

If you are using [Tower](https://www.git-tower.com/) there is a "Sign-Off" checkbox in the commit window. You could even alias git commit to use the `-s` flag so you don’t have to think about it.

By signing your work in this manner, you certify to a "Developer's Certificate of Origin". The current version of this certificate is in the `DCO.txt` file in the root of this repository.

## How-to Guide

There are two ways to make changes, the easy way and the hard way. Either way you will need to [create a GitHub account](https://github.com/signup/free).

Easy way GitHub allows in-line editing of files for making simple typo changes and quick-fixes. This is not the best way as you are unable to test the code works. If you do this you could be introducing syntax errors, etc, but for a Git-phobic user this is good for a quick-fix.

Hard way The best way to contribute is to "clone" your fork of CodeIgniter to your development area. That sounds like some jargon, but "forking" on GitHub means "making a copy of that repo to your account" and "cloning" means "copying that code to your environment so you can work on it".

1. [Set up Git](https://help.github.com/en/articles/set-up-git) (Windows, Mac & Linux)

2. Go to the [CodeIgniter repo](https://github.com/bcit-ci/CodeIgniter)

3. [Fork it](https://help.github.com/en/articles/fork-a-repo)

4. [Clone](https://help.github.com/en/articles/fetching-a-remote#clone) your forked CodeIgniter repo: git@github.com:<your-name>/CodeIgniter.git.

5. Checkout the "develop" branch. At this point you are ready to start making changes.

6. Fix existing bugs on the Issue tracker after taking a look to see nobody else is working on them.

7. [Commit](https://help.github.com/en/articles/adding-a-file-to-a-repository-using-the-command-line) the files

8. [Push](https://help.github.com/en/articles/pushing-to-a-remote) your develop branch to your fork

9. [Send a pull request](https://help.github.com/en/articles/creating-a-pull-request)

The Reactor Engineers will now be alerted about the change and at least one of the team will respond. If your change fails to meet the guidelines it will be bounced, or feedback will be provided to help you improve it.

Once the Reactor Engineer handling your pull request is happy with it they will merge it into develop and your patch will be part of the next release.

### Keeping your fork up-to-date

Unlike systems like Subversion, Git can have multiple remotes. A remote is the name for a URL of a Git repository. By default your fork will have a remote named "origin" which points to your fork, but you can add another remote named "codeigniter" which points to `git://github.com/bcit-ci/CodeIgniter.git`. This is a read-only remote but you can pull from this develop branch to update your own.

If you are using command-line you can do the following:

1. `git remote add codeigniter git://github.com/bcit-ci/CodeIgniter.git`

2. `git pull codeigniter develop`

3. `git push origin develop`

Now your fork is up to date. This should be done regularly, or before you send a pull request at least.