# Contributing to Sming Framework

All contributions (PRs) to Sming Framework have to be done to the \_develop\_ branch.

\_\_master\_\_: Branch that contains latest production (stable) release. No PRs other than Final Release updates will be merged into \_\_master\_\_.

\_\_develop\_\_: Main development branch: contains all current new features.

This means that \_\_all contributors\_\_ will have to submit a PR to a \_develop\_ , it will be tested and then merged to \_\_develop\_\_ for automated integration testing (via TravisCI, Jenkins or SemaphoreCI), as well as manual testing on a real device.

\_\_Sming Contributing flow\_\_:

1. Fork [\_Sming\_ repo](https://github.com/SmingHub/Sming#fork-destination-box)

After that clone your own fork.

`git clone https://github.com/<my-fork>/Sming.git`

2. Create a new branch off the \_develop\_ branch

```

cd Sming

git checkout develop

git branch feature/<short-explanation>

git checkout feature/<short-explanation>

```

Make sure to replace `short-explanation` with one or two words describing the purpose of the feature.

If you want to commit a fix use `fix/` as prefix for your branch name.

3. Build, test your code

Make sure that your code compiles and it is tested on real device. Sometimes you will be asked for a proof (usually screenshot) that it was tested on your device(s).

4. Document your code

As a bare minimum, please include a `README.rst` or `README.md` file. See :doc:`/contribute/documentation` for further information.

5. Commit changes

```

git add <changed-files>

git commit -m "<Description of the change(s)>"

```

6. Push your changes to your fork on github

```

git push

```

7. Rebase if needed

If your branch cannot be merged automatically because there are new changes in the official \_\_develop\_\_

branch that conflict with yours then make sure to rebase your branch. The following steps can help

you do this.

- First step:

You will need to add the `upstream` repository. This step should be executed ONLY once.

```

cd Sming

git remote add upstream https://github.com/SmingHub/Sming.git

git fetch upstream develop

git checkout develop

git reset --hard upstream/develop

```

- Second step:

If you have already defined `upstream` repository and synchronized your `develop` branch to fetch the updates

from `upstream` ( the commands above do this) the next step is to get the latest changes from the official `develop` branch.

This can be done using

```

cd Sming

git checkout develop

git pull

```

- Final step:

Now you are ready to merge the latest changes from official `develop` branch into your branch and place your changes on top.

The commands below help you achieve this.

```

cd Sming

git checkout develop

git pull

git checkout feature/<short-explanation>

git merge develop

# Fix any merge conflicts if needed.

git rebase develop

# Fix any merge conflicts if needed.

```

If there were merge conflicts you will have to resolve them locally. This usually involves calling the following commands:

```

git mergetool

# After resolving conflicts you should type the command below to see what are the next steps

# Newer versions of `git` are kind enough to show hints

git status

```

After that operation you should have a branch that has all latest changes from \_\_develop\_\_

with your changes on top.

8. Submit PR to the main Sming repo, \_\_develop\_\_ branch.

9. Work with other contributors to test your feature and get it merged to \_develop\_

This is the most common approach for a git-flow:

http://nvie.com/posts/a-successful-git-branching-model/