# Contributing guidelines

See the [Docker contributing guidelines](https://github.com/docker/docker/blob/master/CONTRIBUTING.md).

The following is specific to Docker SDK for Python.

Thank you for your interest in the project. We look forward to your

contribution. In order to make the process as fast and streamlined as possible,

here is a set of guidelines we recommend you follow.

## Reporting issues

We do our best to ensure bugs don't creep up in our releases, but some may

still slip through. If you encounter one while using the SDK, please

create an issue

[in the tracker](https://github.com/docker/docker-py/issues/new) with

the following information:

- SDK version, Docker version and python version

```

pip freeze | grep docker && python --version && docker version

```

- OS, distribution and OS version

- The issue you're encountering including a stacktrace if applicable

- If possible, steps or a code snippet to reproduce the issue

To save yourself time, please be sure to check our

[documentation](https://docker-py.readthedocs.io/) and use the

[search function](https://github.com/docker/docker-py/search) to find

out if it has already been addressed, or is currently being looked at.

## Submitting pull requests

Do you have a fix for an existing issue, or want to add a new functionality

to the SDK? We happily welcome pull requests. Here are a few tips to

make the review process easier on both the maintainers and yourself.

### 1. Sign your commits

Please refer to the ["Sign your work"](https://github.com/docker/docker/blob/master/CONTRIBUTING.md#sign-your-work)

paragraph in the Docker contribution guidelines.

### 2. Make sure tests pass

Before we can review your pull request, please ensure that nothing has been

broken by your changes by running the test suite. You can do so simply by

running `make test` in the project root. This also includes coding style using

`flake8`

### 3. Write clear, self-contained commits

Your commit message should be concise and describe the nature of the change.

The commit itself should make sense in isolation from the others in your PR.

Specifically, one should be able to review your commit separately from the

context.

### 4. Rebase proactively

It's much easier to review a pull request that is up to date against the

current master branch.

### 5. Notify thread subscribers when changes are made

GitHub doesn't notify subscribers when new commits happen on a PR, and

fixes or additions might be missed. Please add a comment to the PR thread

when you push new changes.

### 6. Two maintainers LGTM are required for merging

Please wait for review and approval of two maintainers, and respond to their

comments and suggestions during review.

### 7. Add tests

Whether you're adding new functionality to the project or fixing a bug, please

add relevant tests to ensure the code you added continues to work as the

project evolves.

### 8. Add docs

This usually applies to new features rather than bug fixes, but new behavior

should always be documented.

### 9. Ask questions

If you're ever confused about something pertaining to the project, feel free

to reach out and ask questions. We will do our best to answer and help out.

## Development environment

If you're looking contribute to Docker SDK for Python but are new to the

project or Python, here are the steps to get you started.

1. Fork https://github.com/docker/docker-py to your username.

2. Clone your forked repository locally with

`git clone git@github.com:yourusername/docker-py.git`.

3. Configure a

[remote](https://help.github.com/articles/configuring-a-remote-for-a-fork/)

for your fork so that you can

[sync changes you make](https://help.github.com/articles/syncing-a-fork/)

with the original repository.

4. Enter the local directory `cd docker-py`.

5. Run `python setup.py develop` to install the dev version of the project

and required dependencies. We recommend you do so inside a

[virtual environment](http://docs.python-guide.org/en/latest/dev/virtualenvs)

## Running the tests & Code Quality

To get the source source code and run the unit tests, run:

```

$ git clone git://github.com/docker/docker-py.git

$ cd docker-py

$ make test

```

## Building the docs

```

$ make docs

$ open \_build/index.html

```

## Release Checklist

Before a new release, please go through the following checklist:

\* Bump version in docker/version.py

\* Add a release note in docs/change\_log.md

\* Git tag the version

\* Upload to pypi

## Vulnerability Reporting

For any security issues, please do NOT file an issue or pull request on github!

Please contact [security@docker.com](mailto:security@docker.com) or read [the

Docker security page](https://www.docker.com/resources/security/).