# Contributing to Portus

## Reporting issues

Before reporting an issue, please check the

[documentation](http://port.us.org/documentation.html) and our [mailing

list](https://groups.google.com/forum/#!forum/portus-dev). It might contain all

the information you need for your specific problem.

Moreover, check whether it has already been reported

[here](https://github.com/SUSE/Portus/issues). If this is the case, please:

- Read all the comments to confirm that it's the same issue you're having.

- Refrain from adding "same thing here" or "+1" comments. Just hit the

"subscribe" button to get notifications for this issue.

- Add a comment only if you can provide helpful information that has not been

provided in the discussion yet.

## Check for assigned people

We are using Github Issues for submitting known issues (e.g. bugs, features,

etc.). Some issues will have someone assigned, meaning that there's already

someone that takes responsability for fixing said issue. This is not done to

discourage contributions, rather to not step in the work that has already been

done by the assignee. If you want to work on a known issue with someone already

assigned to it, please consider contacting the assignee first (e.g. by

mentioning the assignee in a new comment on the specific issue). This way you

can contribute with ideas, or even with code if the assignee decides that you

can step in.

If you plan to work on a non assigned issue, please add a comment on the issue

to prevent duplicated work.

## Provide tests

In Portus we are \*really\* committed to keep a thorough test suite. For this

reason, any new Pull Request \*always\* has to provide tests for the change

that is being made. The `spec` directory is full of tests that might serve

as an example if you are not sure how to implement tests for your Pull Request.

Moreover, we make use of [Travis-CI](https://travis-ci.org/SUSE/Portus), so we

will only merge your Pull Request once we get a green light from Travis.

You might want to take a look at

[this](https://github.com/SUSE/Portus/wiki/How-we-test-Portus) section from the

wiki where our test infrastructure is more thoroughly explained.

## Mind the Style

We believe that in order to have a healthy codebase we need to abide to a

certain code style. We use [rubocop](https://github.com/bbatsov/rubocop) for

this matter, which is a tool that has proved to be useful. So, before

submitting your Pull Request, make sure that `rubocop` is passing for you.

If you want to know the style we are enforcing, note the following:

- We mainly use the default configuration as stated

[here](https://github.com/bbatsov/rubocop#defaults).

- We've made some small changes to the defaults, as you can see

[here](https://github.com/SUSE/Portus/blob/master/.rubocop.yml). Moreover, note

that all these changes have a comment explaining the reasoning behind it.

Finally, note that `rubocop` is called on Travis-CI. This means that your Pull

Request will not be merged until `rubocop` approves your changes.

## Sign your work

The sign-off is a simple line at the end of the explanation for the patch. Your

signature certifies that you wrote the patch or otherwise have the right to pass

it on as an open-source patch. The rules are pretty simple: if you can certify

the below (from [developercertificate.org](http://developercertificate.org/)):

```

Developer Certificate of Origin

Version 1.1

Copyright (C) 2004, 2006 The Linux Foundation and its contributors.

660 York Street, Suite 102,

San Francisco, CA 94110 USA

Everyone is permitted to copy and distribute verbatim copies of this

license document, but changing it is not allowed.

Developer's Certificate of Origin 1.1

By making a contribution to this project, I certify that:

(a) The contribution was created in whole or in part by me and I

have the right to submit it under the open source license

indicated in the file; or

(b) The contribution is based upon previous work that, to the best

of my knowledge, is covered under an appropriate open source

license and I have the right under that license to submit that

work with modifications, whether created in whole or in part

by me, under the same open source license (unless I am

permitted to submit under a different license), as indicated

in the file; or

(c) The contribution was provided directly to me by some other

person who certified (a), (b) or (c) and I have not modified

it.

(d) I understand and agree that this project and the contribution

are public and that a record of the contribution (including all

personal information I submit with it, including my sign-off) is

maintained indefinitely and may be redistributed consistent with

this project or the open source license(s) involved.

```

Then you just add a line to every git commit message:

Signed-off-by: Joe Smith <joe.smith@email.com>

Use your real name (sorry, no pseudonyms or anonymous contributions.)

If you set your `user.name` and `user.email` git configs, you can sign your

commit automatically with `git commit -s`.