

Commits & Pull-requests Conventions

Commits & Pull-requests Conventions

Commits

In the past, we used to have only one commit by pull request. This was a bad habit. We now require contributors to make **atomic commits**, so you will surely have more than one commit in a single pull request. This will be helpful to review, cherry-pick or revert (we hope to never have to do that) the changes.

What's an atomic commit? It means that the commit's purpose is **one, and only one, complete fix or change**. Typically, ask yourself if what you are doing is one or several tasks. Do not hesitate to use "git add -p ..." (details [here](#)) if you have made several changes in the same file but not all those changes are meant to be in the current commit.

When you are modifying CSS and/or JavaScript files, we invite you to make a separate commit for the compiled files. If you want to know more about compiling assets, you can look at [our documentation](#).

The commit name should also give an idea of the context or of the file that is being changed. The more details, the better! The commit name should be as unique and recognizable as your commit itself.

To make the commits context easier to read and hassle-free, we have chosen to prefix them with two letters depending on the modification scope:

Code	Scope
FO	if the changes impact the front office
BO	if the changes impact the back office
IN	if the changes impact the installer
WS	if the changes impact the web service
CO	if the changes impact the core
TE	if the changes impact the tests

Examples:

- CO: Retrieve all required data for notifications
- BO: JS for notifications in new BO theme
- FO: Fix voucher layout in cart
- TE: PHP 7 is no longer an option

Pull Requests

Now that you have made atomic commits, you surely have a lot of commits for one pull request. A pull request answers to a given issue. Do not ever make a single pull request for many purposes. Do not hesitate to split your big commit into several subprojects. It will be easier and quicker to review.

As all your commits messages are well-formated, just make a summary of your pull request's purpose in its GitHub title. A summary does not mean it can not be explicit. Please describe what your pull request does in details (avoid "Fix product page" or "Category page improvement"). Then, just fill the PR template table to answer some questions which will help the team make a decision faster.

Please note that all the pull requests **must** follow those guidelines. If the commit messages are not well-formatted, the pull request's title is not correct, or the table is not properly filled, we will not be able to accept your pull request.

Find more information and all the details on [our documentation](#).