

Contributing

We welcome contributions to all projects of FenixEdu. If you have something that you consider useful to a project, read this carefully to understand how to submit that, and what you can expect from the evaluators.

JIRA Discussion

The first step is to discuss your issue in the project's JIRA issue tracker. Each project in the family has a separate issue tracking, the current list of projects is:

Project	Issue Tracker	Description	Project Lead
 FenixEdu Academic	ACDM	Academic Administration (SIS)	Sérgio Silva
 FenixEdu Learning	LRN	Learning Management System	Sérgio Silva
 FenixEdu Spaces	SPC	Space Management and Allocation	Sérgio Silva
 FenixEdu CMS	CMS	Content Management System	Sérgio Silva
 FenixEdu Canvas	CNV	CSS Bootstrap Based Framework	Rita Severo
 FenixEdu Academic Thesis	THS	Thesis Management Addon for Academic	Sérgio Silva
 Bennu	BNN	Web Tier over the fenix-framework	David Martinho
 Bennu Renderers	BRND	Renderers UI Backend for Bennu	David Martinho
 FenixEdu Commons	CMN	Commons Library	David Martinho
 Oddjet	ODJ	Document Templating	David Martinho

Start by searching, it might be the case that already is an issue that addresses the same concern. If you do not find anything, then create a new JIRA issue. Trivial bug fixes, typo fixes, code cleanup can be immediately followed by the pull-request. Anything over that will be thoroughly discussed before even be considered by the maintainers.

Contribute Your Changeset

Create a Branch

Branch from the 'master' of the project you are submitting to. The 'master' branch contains the code in the current development cycle and it's where most contributions will be accepted. In rarer cases backports will be accepted to a maintenance branch (the ones like 4.1.x), in which case the branch would have been made from that maintenance branch.

The branch name if of your choosing but a good practice would be to prefix with the type of submission: fix, newfeature, refactor, enhancement followed by a slash '/', then the issue key, or a short issue lower case description. Either way make them short and expressive.

Keep Projects Formatting and Style

All projects follow the rules stated [here](#). Follow these conventions and at least your java submissions will be restricted to actual change set. Every other language is not mandated by any strong set of rules, so just keep it like it is. Submissions where the contribution is mixed with a lot of white space changes will be rejected, see the git diff --check command to check for whitespace errors. Besides that, don't try to do things your way, adjust yourself to the existing style.

License and Authorship

All new files should be prefixed with the license header, just copy paste from an existing file, and don't change the copyright line. The javadoc may contain an @author reference.

Prepare Your Commit

We expect a single commit, per pull request, per issue. If during your development you ended up with several, then squash them before submitting. Use your real name in the commit.

Format your commit messages, as a general rule, your messages should start with a single line that's no more than about 50 characters and that describes the changeset concisely, followed by a blank line, followed by a more detailed explanation. Write proper english, and case properly, use imperative sentences, keep it short but complete.

Submit Your Pull Request

The subject line should follow the same conventions as the commit's subject line. Optionally include [JIRA Smart Commits](#) to act on the issue upon acceptance, e.g. `ACDM-123 #resolve`. In any case the JIRA issue must be mentioned. On the JIRA side its likely that the issue is linked to the pull request, particularly if you used JIRA Smart Commits, if not be sure to link it yourself. We look for two-way traceability.

Expect Discussion and Rework

Code quality and stability is an huge issue, as these projects tend to gain an also huge complexity over time. Maintainer work to keep this complexity down, so expect to be asked to rework your changeset. You can force push reworked or rebased commits at any time. Note that, the further the issue is discussed prior to code submission, the more chances you get to be accepted with minimum work.