Contribution Guidelines

Magnolia's code lives under several Git repositories at https://git.magnolia-cms.com.
We use Bitbucket's pull requests for code reviews and 3rd-party contributions.

Report an issue

The first step to contribute to Magnolia is a Jira ticket for the particular contribution (*e.g.* bug fix, improvement, feature).

> ✅ You might also want to consider contacting Magnolia Support as a first step. Support can help with facilitating your contribution into the PD pipeline. They can also help find workarounds.

🔔 Do search for pre-existing tickets beforehand to avoid duplicates.

How to create a Pull Request

1. **Create a fork** of the module repository you want to submit a change to.

2. **Create a development branch** on this fork; check out that branch locally.

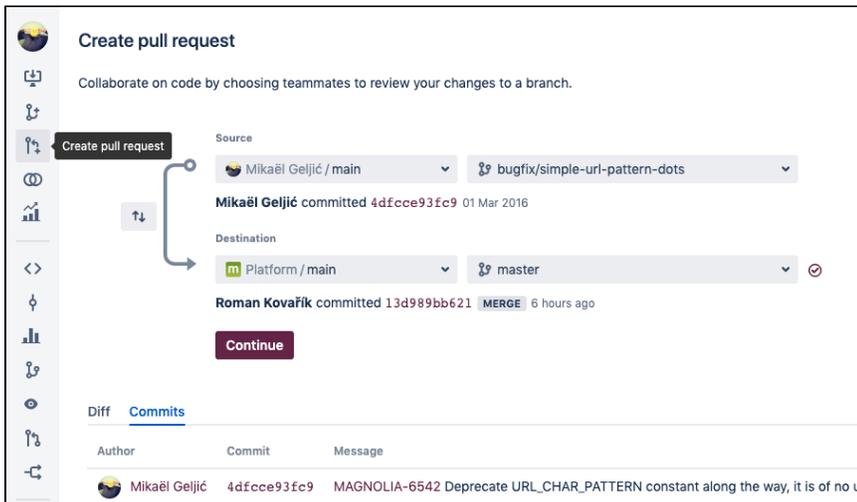🔔 Conveniently, Jira has a *Create branch* link for that.

3. **Commit and push** to that branch.

- Make atomic commits, that is incremental, ever compiling changes.
- Commit behavior changes separately from cosmetic changes such as variable renames, method reordering, formatting.
- Prefix behavior-changing commit messages with the Jira ticket number.
- Do read conventional commit message guidelines for additional details.

🔔 If you push from the Terminal, Bitbucket suggests you the URL to create the pull request immediately.

4. **Create the pull-request**

- Make sure to select Magnolia's corresponding "canonical repository" in *Destination* (those in projects such as *Platform*, *Modules*, etc.);
- Destination branch should be `master` in most cases

5. **Tell us** about it!

If you have an enterprise subscription, advise your support contact at Magnolia, to make sure we assign a Magnolia staff member for review.
If you are a community user, mind that we groom community tickets in Jira once a week, so please create a ticket in Jira describing your fix or improvement and mentioning that there is a PR. You may still drop us a line on the Magnolia Developers Mailing List 😄.

Once assigned, Magnolia developers will help you as best as possible to discuss the changes, as well as to pass various validation steps (*e.g.* code style, tests, PR builds).