Git workflow

How Git is used for Exponential - main as the one development line, branches and pull requests squashed and rebased while open, merge commits, no history rewriting after a merge, commit subjects with Added, Updated…

Source file: `doc/eng-us/help/software-engineers/git-workflow.md` Last changed 2026-11-04
https://doc.exponential.earth/learninminutes/exphelp/viewer/help/software-engineers/git-workflow.md

This page is for engineers who commit to Exponential, review its history or run installations made of Git clones. The standard is Source control and releases; the hands-on guides are Develop with Git and Update with Git.

Contents

1. Branches and merges
2. Commit messages
3. Hooks
4. What never goes into a commit
5. Installations made of clones
6. Known gaps

Branches and merges

- `main` **is the development line**, aliased `6.0.x-dev` for Composer; CI runs on it.
- **Work happens on branches** and reaches `main` by pull request.
- **While a pull request is open**, its author squashes it into logical commits (usually one) and rebases it onto `main` as the reviews ask:

  Shell

  ```
  git fetch origin
  git rebase -i origin/main
  git push --force-with-lease
  ```

  Force-push only the branch of your own pull request.
- **Pull requests are merged with a merge commit.** Once merged, `main`'s history is never rewritten; later fixes come as new pull requests.

Commit messages

The subject starts with what the change does:

Prefix	For
`Added:`	a new feature, file, class, setting or capability
`Updated:`	a change to existing behavior, templates, settings or code

Prefix	For
`Fixed:`	a bug fix
`Removed:`	something deleted
`Renamed:`	a renamed file, class, setting or extension

Example from the hook: `Fixed: The redirect keeps the query string on the own host`. Git's own `Merge`, `Revert`, `fixup!`, `squash!` and `amend!` subjects pass.

The body says why and what, for a reader who does not know the context. Further rules, not checked by any tool: name the product Exponential (identifiers such as `eZContentObject` stay), no attribution to tools or assistants, no private or unpublished paths, and do not combine change types in one commit when they can be split.

A project author's bad subject fails the check; a contributor's only warns. To check your branch strictly:

Shell

```
EXP_COMMIT_MSG_STRICT_ALL=1 sh .githooks/check-commit-range origin/main HEAD
```

Hooks

Shell

```
make hooks
```

sets `core.hooksPath` to `.githooks`: the pre-commit hook checks syntax (also with PHP 8.0 when found), the file manifest and the code style of staged files; the commit-msg hook checks the subject; the pre-push hook runs PHPStan when the development tools are installed. Skip them only in an emergency (`--no-verify`); CI runs the same checks (Static analysis and code style).

What never goes into a commit

`.gitignore` already excludes much of it; check anyway:

- credentials and machine settings: `settings/override/`, `config.env.php`;
- runtime directories: `var/cache/`, `var/log/`, `var/storage/`;
- `vendor/`, `composer.lock`, `.devtools/`;
- private working notes.

Stage files by name rather than with `git add -A`, so nothing untracked slips in.

Installations made of clones

Composer can install the kernel and the extensions as Git clones. Then you can see what changed, find when and why a line changed, bisect a regression, use a fix before it is released, and keep up to date with a merge (Develop with Git). Updating such an installation is a seven-step procedure with a maintenance window, from the command line or from the administration (Update with Git).

Known gaps

The merge and push rules are policy: `main` has no branch protection rule, and commits and tags are not signed.

Related pages

- Contributing upstream
- Commit messages
- Running a team on Git
- Free documentation: this page is free to copy, translate and improve under the GNU FDL 1.3.

> **Become an Exponential Partner**
>
> Partners share their information, provide funding, contribute back and develop the products with us, and that is how Exponential grows. A partner website and dashboard are planned for mid-2027: please become a partner today to speed up our growth rate and land new features faster.
>
> Partners: start here | How to apply

Exponential Copyright © 1998-2026 Exponential Foundation and 7x
Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".
This documentation is free documentation, licensed under the GNU Free Documentation License 1.3 or later:
https://www.gnu.org/licenses/fdl-1.3.html.