

# AGit Workflow Usage

---

Forgejo ships with limited support for [AGit-Flow](#).

Similarly to [Gerrit's workflow](#), this workflow provides a way of submitting changes to repositories hosted on Forgejo instances using the `git push` command alone, without having to create forks or feature branches and then using the web UI to create a pull request.

Using Push Options (`-o`) and a [Refspec](#) (a location identifier known to Git), it is possible to supply the information required to open a pull request, such as the target branch or the pull request's title.

## Creating pull requests <#>

For clarity, this document will start with some examples.

A full list of the parameters, as well as information on avoiding duplicate pull requests when rebasing or amending commits, will follow.

## Usage examples <#>

Suppose that you cloned a repository and created a new commit on top of the `main` branch. A pull request targeting the `main` branch using your **currently checked out branch** can be created like this:

```
git push origin HEAD:refs/for/main -o topic="agit-typo-fixes"
```

Note that `HEAD:refs/for/main` is the [Refspec](#). `HEAD` refers to the [checked out reference](#), but can be replaced with any ["local ref"](#) (e.g. branch). `refs/for/main` refers to the destination (**"remote ref"**), with `main` being the **"target branch"**, as in the branch that your submitted change should be applied to.

The topic will be visible in the pull request and it will be used to associate further commits with the same pull request.

### Setting a topic using the 'topic' parameter. #

Topics can also be defined using the `<topic>` parameter in the **Refspec**. The following example does the same as the example above: Create a new pull request using the currently checked out reference (**HEAD**) with target branch **main** and topic **agit-typo-fixes**.

```
# `agit-typo-fixes` is the topic parameter
git push origin HEAD:refs/for/main/agit-typo-fixes
```

### Pushing a non-checked-out reference (non-HEAD) #

Suppose you would like to submit a pull request meant for a remote branch called **new-feature** using topic **implement-part12**. However, the changes that you want to submit reside in a local branch called **my-new-feature** that you have not checked out. In order to submit the changes residing in the **my-new-feature** branch **without** checking it out, you can supply the name of the local branch (**my-new-feature**) as follows:

```
git push origin my-new-feature:refs/for/new-feature/implement-part12
```

### Setting a title and a description in AGit #

It is also possible to use some additional push options, such as **title** and **description**. Here's another example targeting the **main** branch:

```
git push origin HEAD:refs/for/main -o topic="implement-part12" \
-o title="Title of the PR" \
-o description="This can be any markdown content."
```

Both push options are optional. If no **title** or **description** push option is included, the title and the description of the first commit will be used in their place instead.

New lines are not supported in the **description** push option. Multi-line descriptions can be encoded in base64 to work around this. (See [forgejo/forgejo#8479](#))

### Changing the default push method #

To push commits to your pull request without having to specify the Refspec, you can modify the default push method to **upstream** in your Git configuration:

```
# To only set this option for this specific repository
git config push.default upstream
```

Then, run the following command:

```
git config branch.my-new-feature.merge refs/for/main/implement-part12
```

After doing so, you can now simply run **git push** to push commits to your pull request, without having to specify the refspec. This will also allow you to pull, fetch, rebase, etc., from the AGit pull request by default.

## Parameters #

```
git push <remote-name> <local-ref>:refs/<for|draft|for-review>/<branch>/<topic> [-o <top
```

The following parameters are available:

- **<remote-name>**: The name of the remote repository (e.g., **origin**) **(required)**
- **<local-ref>**: The local reference being pushed (e.g., **HEAD**, **my-branch**, a commit hash) **(required)**
- **refs/<for|draft|for-review>/<branch>/<topic>**: Refspec **(required)**
  - **for**, **draft**, **for-review**: This parameter describes the pull request type. **for** opens a normal pull request. **draft** and **for-review** are currently silently ignored.
  - **<branch>**: The target branch that a pull request should be merged against **(required)**
  - **<topic>**: The topic for the remote pull request. **If left empty**, the topic must be supplied using the **-o topic** option.
- **-o <topic|title|description|force-push>**: Push options
  - **topic**: Essentially an identifier. **If left empty**, the value of **<topic>**, if present, will be used for the topic. Otherwise, Forgejo will return an error.
  - **title**: Title of the pull request. **If left empty**, the first line of the first new Git commit will be used instead.

- **description:** Description of the pull request. **If left empty**, the description of the first new Git commit will be used instead.
- **force-push:** Necessary when rebasing, amending, or retroactively modifying your previous commits. Otherwise, a new pull request will be opened, **even if you use the same topic**. If used, the value of this parameter should be set to **true**.

If you want to push additional commits to a pull request that was created using AGit, you **must** use the same topic. Forgejo relies on the **topic** parameter and a linear commit history in order to associate new commits with an existing open pull request. Should you wish to overwrite the contents of an existing pull request, use the **force-push** parameter.

*“**Note:** A new pull request will be created if the **topic** was used in a pull request that is merged or closed, independently of the **force-push** parameter.”*

**For Gerrit users:** Forgejo does not support Gerrit’s Change-Ids.

## Review Workflow #

The web UI can be used to review AGit pull requests like any other.

To get command line instructions, click the “View command line instructions” drop-down at the bottom of the pull request form.

 This page's content is available under the Apache-2.0 license. It is derived from this original source.

 [Edit this page](#)

### Forgejo

Releases

News

Fediverse (Mastodon)

Chat room (Matrix)

RSS feed

### Community

Code

Governance

Teams

Identity proofs

### Contribute

Issue tracker

Developer Matrix room

Contributor guide

Localization guide

Donate via Liberapay

### Resources

### Legal

[Status \(Forgejo\)](#)

[Imprint](#)

[Status \(Codeberg\)](#)

[Privacy Policy](#)

[Documentation](#)

[Code of Conduct](#)

[Delightful Forgejo](#)

[Professional services](#)

Copyright © 2026 Forgejo authors. Content available under [CC BY-SA 4.0](#), unless stated otherwise.

[Forgejo mascot](#) by [David Revoy](#), [CC BY 4.0](#).

