

Merging a pull request

In this article

About pull request merges

Merging a pull request

Further reading

Merge a pull request into the upstream branch when work is completed. Anyone with push access to the repository can complete the merge.

[GitHub CLI](#)
[Web browser](#)

About pull request merges

In a pull request, you propose that changes you've made on a head branch should be merged into a base branch. By default, any pull request can be merged at any time, unless the head branch is in conflict with the base branch. However, there may be restrictions on when you can merge a pull request into a specific branch. For example, you may only be able to merge a pull request into the default branch if required status checks are passing. Repository administrators can add constraints like this to branches using branch protection rules or repository rulesets. For more information, see "[About protected branches](#)" and "[About rulesets](#)."

You can configure a pull request to merge automatically when all merge requirements are met. For more information, see "[Automatically merging a pull request](#)."

If the pull request has merge conflicts, or if you'd like to test the changes before merging, you can [check out the pull request locally](#) and merge it using the command line.

You can't merge a draft pull request. For more information about draft pull requests, see "[About pull requests](#)."

The repository may be configured so that the head branch for a pull request is automatically deleted when you merge a pull request. For more information, see "[Managing the automatic deletion of branches](#)."

Note: If you delete a head branch after its pull request has been merged, GitHub checks for any open pull requests in the same repository that specify the deleted branch as their base branch. GitHub automatically updates any such pull requests, changing their base branch to the merged pull request's base branch. For more information, see "[About branches](#)."

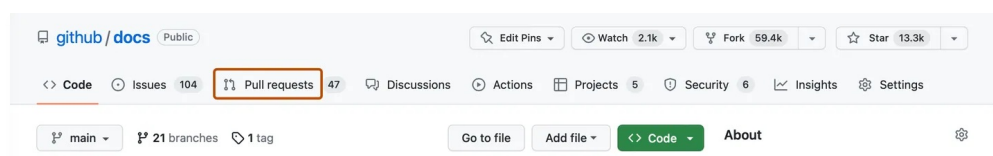
Pull requests are merged using [the `--no-ff` option](#), except for [pull requests with squashed or rebased commits](#), which are merged using the fast-forward option.

You can link a pull request to an issue to show that a fix is in progress and to automatically close the issue when someone merges the pull request. For more information, see "[Linking a pull request to an issue](#)."

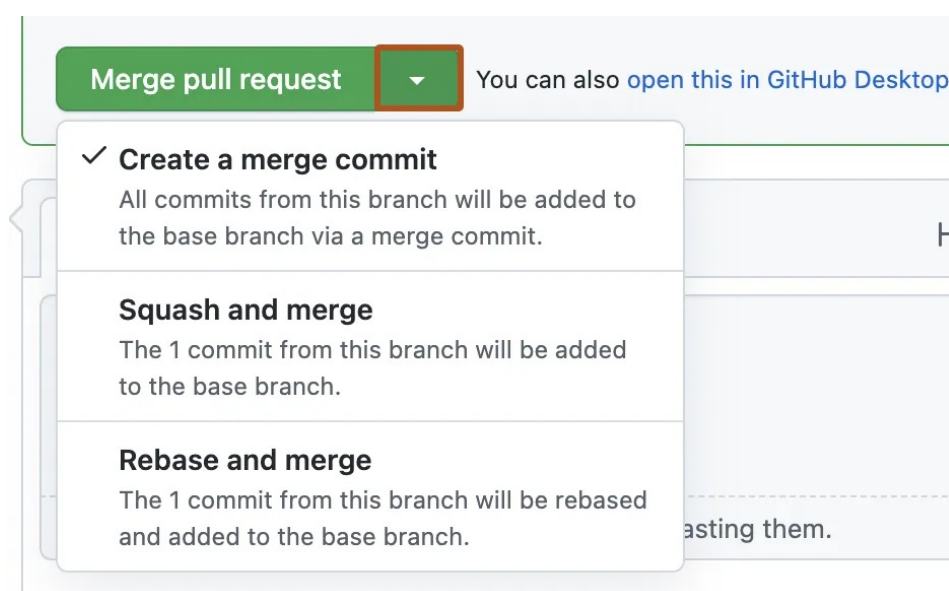
If you decide you don't want the changes in a topic branch to be merged to the upstream branch, you can [close the pull request](#) without merging.

Merging a pull request [↗](#)

- 1 Under your repository name, click **Pull requests**.



- 2 In the "Pull Requests" list, click the pull request you'd like to merge.
- 3 Scroll down to the bottom of the pull request. Depending on the merge options enabled for your repository, you can:
 - [Merge all of the commits into the base branch](#) by clicking **Merge pull request**. If the **Merge pull request** option is not shown, click the merge dropdown menu and select **Create a merge commit**.



- [Squash the commits into one commit](#) by clicking the merge dropdown menu, selecting **Squash and merge** and then clicking **Squash and merge**.
- [Rebase the commits individually onto the base branch](#) by clicking the merge dropdown menu, selecting **Rebase and merge** and then clicking **Rebase and merge**.

Note: Rebase and merge will always update the committer information and create new commit SHAs. For more information, see "[About pull request merges](#)."

- 4 If prompted, type a commit message, or accept the default message.


For information about the default commit messages for squash merges, see "[About pull request merges](#)."

- 5 If you have more than one email address associated with your account on GitHub.com, click the email address drop-down menu and select the email address to use as the Git author email address. Only verified email addresses appear in this drop-down menu. If you enabled email address privacy, then a no-reply will be the

default commit author email address. For more information about the exact form the no-reply email address can take, see "[Setting your commit email address](#)."

Update README.md

Add an optional extended description...

 ☒ octocat@github.com
☐ octocat@github.com

Note: The email selector is not available for rebase merges, which do not create a merge commit. For squash merges, the email selector is only shown if you are the pull request author and you have more than one email address associated with your account.

- 6 Click **Confirm merge**, **Confirm squash and merge**, or **Confirm rebase and merge**.
- 7 Optionally, [delete the branch](#). This keeps the list of branches in your repository tidy.

To learn more about GitHub CLI, see "[About GitHub CLI](#)."

To merge a pull request, use the `gh pr merge` subcommand. Replace `pull-request` with the number, URL, or head branch of the pull request.

```
gh pr merge PULL-REQUEST
```

Follow the interactive prompts to complete the merge. For more information about the merge methods that you can choose, see "[About pull request merges](#)."

Alternatively, you can use flags to skip the interactive prompts. For example, this command will squash the commits into a single commit with the commit message "my squash commit", merge the squashed commit into the base branch, and then delete the local and remote branch.

```
gh pr merge 523 --squash --body "my squash commit" --delete-branch
```

Further reading [↗](#)

- "[Reverting a pull request](#)"
- "[Syncing your branch in GitHub Desktop](#)" using GitHub Desktop
- "[About pull request merges](#)"
- "[Addressing merge conflicts](#)"

Legal

© 2023 GitHub, Inc. [Terms](#) [Privacy](#) [Status](#) [Pricing](#) [Expert services](#) [Blog](#)