


# Resolving a merge conflict on GitHub

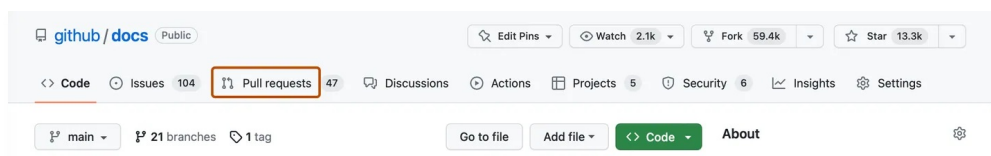
You can resolve simple merge conflicts that involve competing line changes on GitHub, using the conflict editor.

You can only resolve merge conflicts on GitHub Enterprise Server that are caused by competing line changes, such as when people make different changes to the same line of the same file on different branches in your Git repository. For all other types of merge conflicts, you must resolve the conflict locally on the command line. For more information, see "[Resolving a merge conflict using the command line](#)."

If a site administrator disables the merge conflict editor for pull requests between repositories, you cannot use the conflict editor on GitHub Enterprise Server and must resolve merge conflicts on the command line. For example, if the merge conflict editor is disabled, you cannot use it on a pull request between a fork and upstream repository.

**Warning:** When you resolve a merge conflict on GitHub Enterprise Server, the entire [base branch](#) of your pull request is merged into the [head branch](#). Make sure you really want to commit to this branch. If the head branch is the default branch of your repository, you'll be given the option of creating a new branch to serve as the head branch for your pull request. If the head branch is protected you won't be able to merge your conflict resolution into it, so you'll be prompted to create a new head branch. For more information, see "[About protected branches](#)."

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



- 2 In the "Pull Requests" list, click the pull request with a merge conflict that you'd like to resolve.
- 3 Near the bottom of your pull request, click **Resolve conflicts**.



**This branch has conflicts that must be resolved**

Use the [web editor](#) or the [command line](#) to resolve conflicts.

**Conflicting files**

content/issues/tracking-your-work-with-issues/linking-a-pull-request-to-an-issue.md

Resolve conflicts

**Note:** If the **Resolve conflicts** button is deactivated, your pull request's merge conflict is too complex to resolve on GitHub Enterprise Server or the site administrator has disabled the conflict editor for pull requests between repositories. You must resolve the merge conflict using an alternative Git client, or by using Git on the command line. For more information see "[Resolving a merge conflict using the command line](#)."

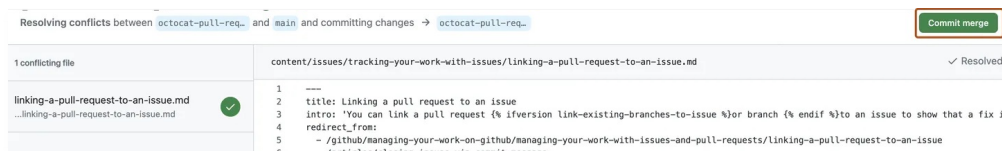
- 4 Decide if you want to keep only your branch's changes, keep only the other branch's changes, or make a brand new change, which may incorporate changes

from both branches. Delete the conflict markers `<<<<<<<`, `=====`, `>>>>>>>` and make the changes you want in the final merge.

- 5 If you have more than one merge conflict in your file, scroll down to the next set of conflict markers and repeat steps four and five to resolve your merge conflict.
- 6 Once you've resolved all the conflicts in the file, click **Mark as resolved**.



- 7 If you have more than one file with a conflict, select the next file you want to edit on the left side of the page under "conflicting files" and repeat steps four through seven until you've resolved all of your pull request's merge conflicts.
- 8 Once you've resolved all your merge conflicts, click **Commit merge**. This merges the entire base branch into your head branch.



- 9 If prompted, review the branch that you are committing to.

If the head branch is the default branch of the repository, you can choose either to update this branch with the changes you made to resolve the conflict, or to create a new branch and use this as the head branch of the pull request.

If you choose to create a new branch, enter a name for the branch.

If the head branch of your pull request is protected you must create a new branch. You won't get the option to update the protected branch.

Click **Create branch and update my pull request** or **I understand, continue updating BRANCH**. The button text corresponds to the action you are performing.

- 10 To merge your pull request, click **Merge pull request**. For more information about other pull request merge options, see "[Merging a pull request](#)."

## Further reading [↗](#)

- "[About pull request merges](#)"

## Legal

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