Home

Pull requests

- Commit changes to your project
- Collaborate with pull requests
  - Getting started
  - Working with forks
  - Code quality features
  - Propose changes
  - Address merge conflicts
    - About merge conflicts
    - Resolve merge conflicts
    - Resolve merge conflicts in Git
  - Review changes
  - Incorporate changes

Pull requests / Collaborate with pull requests / Address merge conflicts /

About merge conflicts

Merge conflicts happen when you merge branches that have competing commits, and Git needs your help to decide which changes to incorporate in the final merge.

Git can often resolve differences between branches and merge them automatically. Usually, the changes are on different lines, or even in different files, which makes the merge simple for computers to understand. However, sometimes there are competing changes that Git can't resolve without your help. Often, merge conflicts happen when people make different changes to the same line of the same file, or when one person edits a file and another person deletes the same file.

You must resolve all merge conflicts before you can merge a pull request on GitHub. If you have a merge conflict between the compare branch and base branch in your pull request, you can view a list of the files with conflicting changes above the **Merge pull request** button. The **Merge pull request** button is deactivated until you've resolved all conflicts between the compare branch and base branch.

Resolving merge conflicts

To resolve a merge conflict, you must manually edit the conflicted file to select the changes that you want to keep in the final merge. There are a couple of different ways to resolve a merge conflict:

- If your merge conflict is 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, you can resolve it on GitHub using the conflict editor. For more information, see "Resolving a merge conflict on GitHub."
- For all other types of merge conflicts, you must resolve the merge conflict in a local clone of the repository and push the change to your branch on GitHub. You can use the command line or a tool like GitHub Desktop to push the change. For more information, see "Resolving a merge conflict using the command line."

If you have a merge conflict on the command line, you cannot push your local changes to GitHub until you resolve the merge conflict locally on your computer. If you try merging branches on the command line that have a merge conflict, you'll get an error message. For more information, see "Resolving a merge conflict using the command line."

```
$ git merge BRANCH-NAME
> Auto-merging styleguide.md
> CONFLICT (content): Merge conflict in styleguide.md
> Automatic merge failed; fix conflicts and then commit the result
```

Further reading

- "About pull request merges"
- "About pull requests"
- "Resolving a merge conflict using the command line"
- "Resolving a merge conflict on GitHub"

Help and support

Did this doc help you?

Privacy policy

Help us make these docs great!

All GitHub docs are open source. See something that's wrong or unclear? Submit a pull request.

Make a contribution

Learn how to contribute

Still need help?

Ask the GitHub community

Contact support