Syncing your branch in GitHub Desktop

As commits are pushed to your project on GitHub, you can keep your local copy of the project in sync by pulling from the remote repository.

Mac   Windows

About branch synchronization 🔗

You can sync your local branch with the remote repository by pulling any commits that have been added to the branch on GitHub since the last time you synced. If you make commits from another device or if multiple people contribute to a project, you will need to sync your local branch to keep the branch updated.
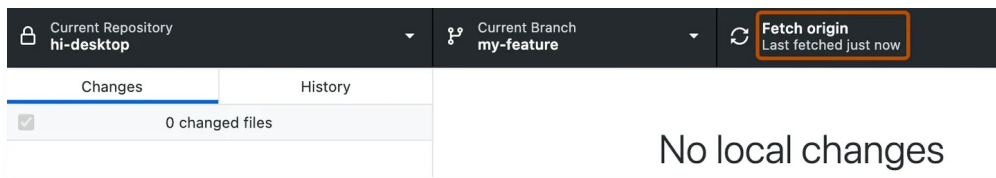
When you pull to your local branch, you only update your local copy of the repository. To update your branch on GitHub, you must push your changes. For more information, see "Pushing changes to GitHub from GitHub Desktop."

To add changes from one branch to another branch, you can merge the branches. To apply changes to your branch from another branch in the same repository, you can merge the other branch into your branch on GitHub Desktop. To request that changes from your branch are merged into another branch, in the same repository or in another repository in the network, you can create a pull request on GitHub Desktop. For more information, see "Merging another branch into your project branch" and "About pull requests."

Some workflows require or benefit from rebasing instead of merging. By rebasing you can reorder, edit, or squash commits together. For more information, see "About Git rebase" and "Rebasing your project branch onto another branch."

Pulling to your local branch from the remote 🔗

1 In GitHub Desktop, use the ⎇ **Current Branch** drop-down, and select the local branch you want to update.

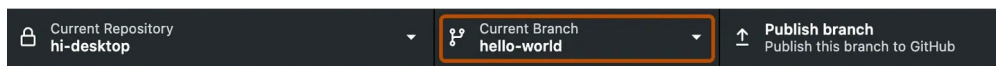2 To check for commits on the remote branch, click **Fetch origin**

🔒 Current Repository hi-desktop ▾	ᛦ Current Branch my-feature ▾	⟳ Fetch origin Last fetched just now

Changes	History

☑ 0 changed files

No local changes

③ To pull any commits from the remote branch, click **Pull origin** or **Pull origin with rebase**.

🖥 Current Repository docs ▾	ᛦ Current Branch main ▾	↓ Pull origin Last fetched just now 4 ↓

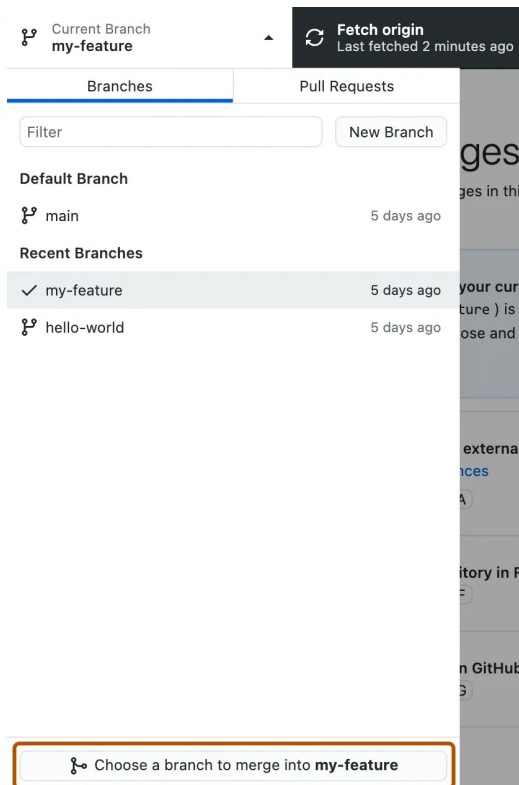| Changes	History

☑ 0 changed files

No local changes

④ Resolve any merge conflicts in your preferred way, using a text editor, the command line, or another tool. For more information, see "Addressing merge conflicts."

Merging another branch into your project branch 🔗
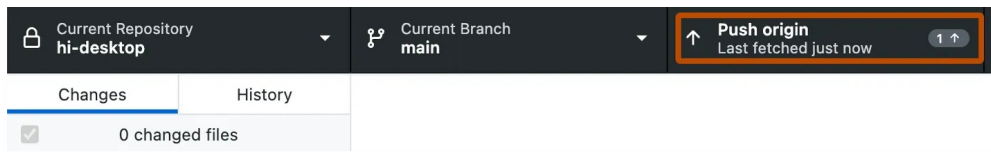
① In GitHub Desktop, click ᛦ **Current Branch**.

🔒 Current Repository hi-desktop ▾	ᛦ Current Branch hello-world ▾	↑ Publish branch Publish this branch to GitHub

② Click **Choose a branch to merge into BRANCH**.

ᛦ Current Branch **my-feature** ▴   ⟳ Fetch origin Last fetched 2 minutes ago

Branches	Pull Requests

Filter   New Branch

Default Branch

ᛦ main   5 days ago

Recent Branches

✓ my-feature   5 days ago

ᛦ hello-world   5 days ago

ᛦ Choose a branch to merge into **my-feature**

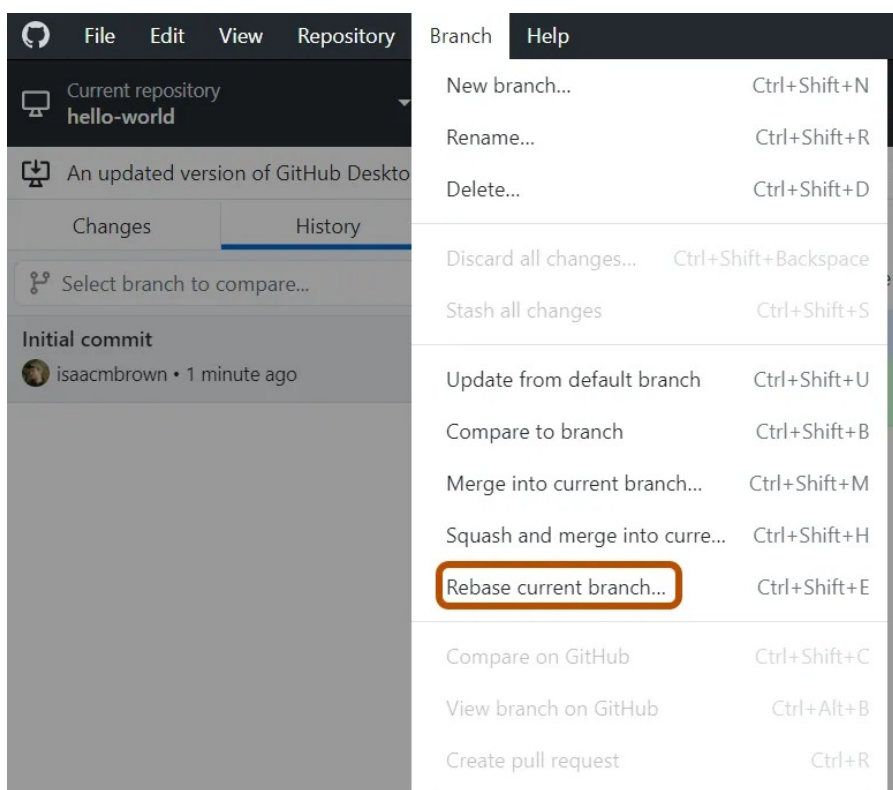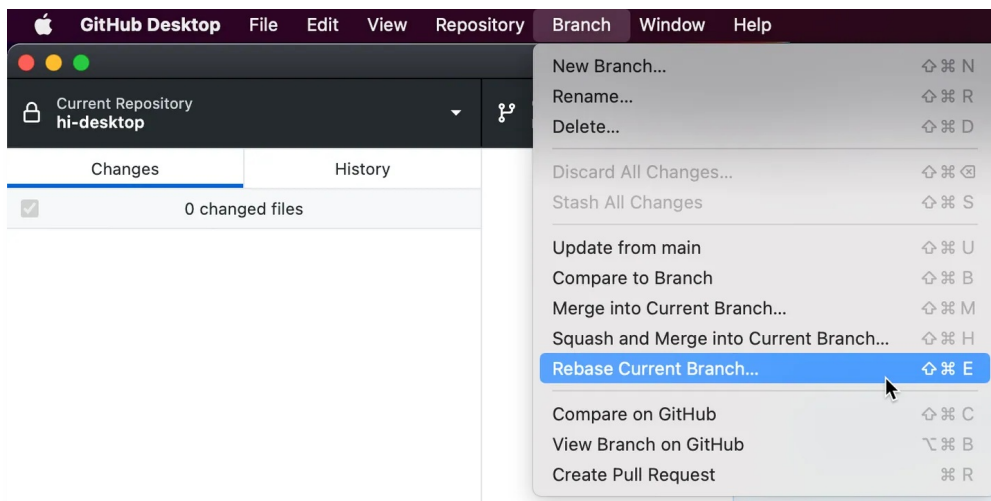③ Click the branch you want to merge into the current branch, then click **Merge BRANCH into BRANCH**.

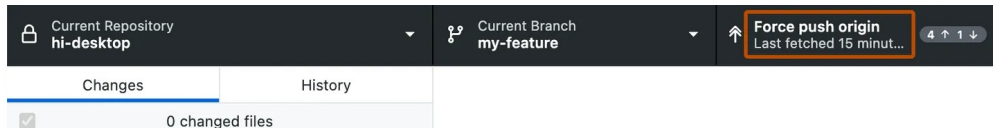4   To push your local changes to the remote repository, in the repository bar, click **Push origin**.

Rebasing your project branch onto another branch 🔗

1   In the menu bar, select **Branch**, then click **Rebase Current Branch**.

2　Click the branch you want to rebase into the current branch, then click **Rebase**.

3　If you're sure you want to rebase, click **Begin Rebase**.

4　Resolve any merge conflicts in your preferred way, using a text editor, the command line, or another tool. For more information, see "Addressing merge conflicts."

5　To push your local changes, click **Force push origin**.

Squashing and merging another branch into your project branch 🔗

1　In the menu bar, select **Branch**, then click **Squash and Merge into Current Branch**.
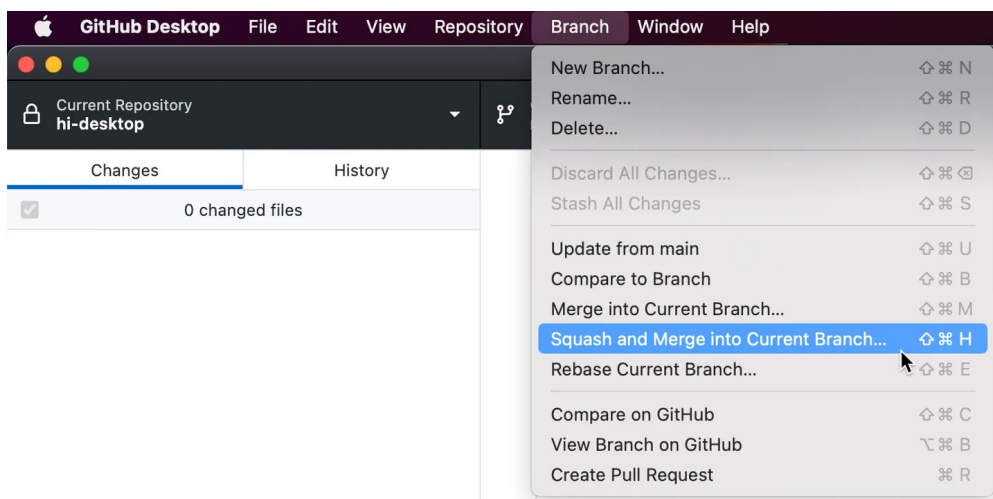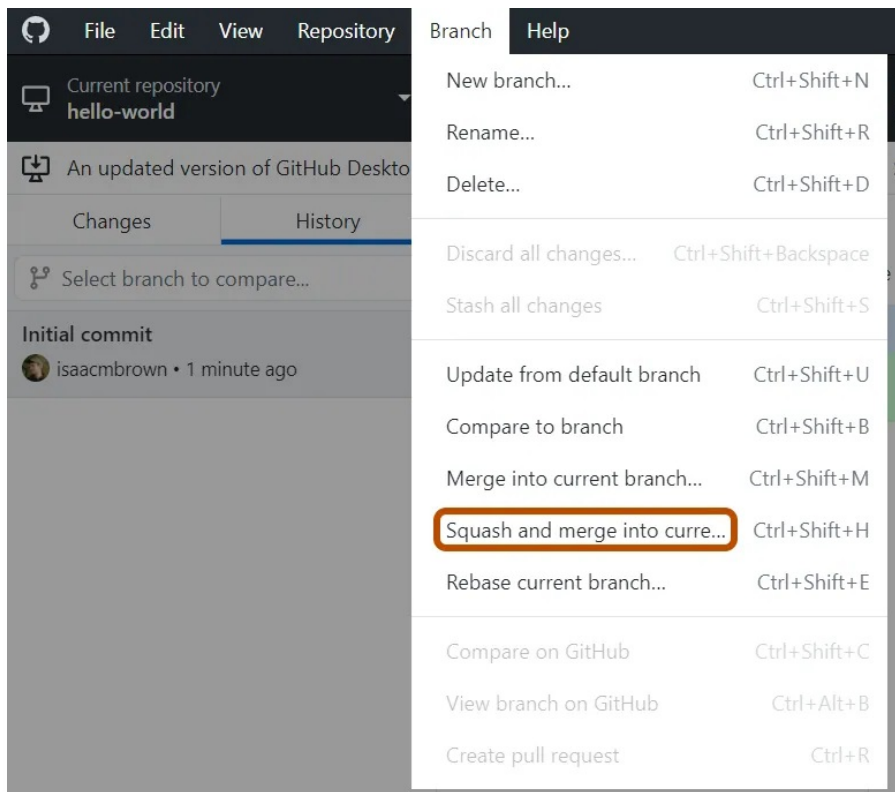
2 In the "Squash and merge" window, click the branch you want to merge into the current branch, then click **Squash and merge**.

> **Note:** If there are merge conflicts, GitHub Desktop will warn you above the **Squash and merge** button. You will not be able to squash and merge the branch until you have resolved all conflicts.

3 To push your local changes to the remote repository, in the repository bar, click **Push origin**.
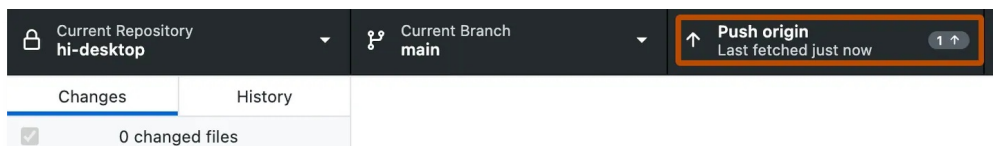
Further Reading 🔗

- "[GitHub glossary](#)" in the GitHub glossary
- "[GitHub glossary](#)" in the GitHub glossary
- "[GitHub glossary](#)" in the GitHub glossary