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
Review changes
About PR reviews
Review proposed changes
Filter files
Methods & functions
Comment on a PR
View a PR review
Review dependency changes
Incorporate feedback

Reviewing proposed changes in a pull request

In a pull request, you can review and discuss commits, changed files, and the differences (or "diff") between the files in the base and compare branches.

Codespaces | Web browser

In this article
About reviewing pull requests
Starting a review
Reviewing dependency changes
Marking a file as viewed
Submitting your review
Further reading

About reviewing pull requests

You can review changes in a pull request one file at a time. While reviewing the files in a pull request, you can leave individual comments on specific changes. After you finish reviewing each file, you can mark the file as viewed. This collapses the file, helping you identify the files you still need to review. A progress bar in the pull request header shows the number of files you've viewed. After reviewing as many files as you want, you can approve the pull request or request additional changes by submitting your review with a summary comment.

Tip: You can find a pull request where you or a team you're a member of is requested for review with the search qualifier `review-requested:[USERNAME]` or `team-review-requested:[TEAMNAME]`. For more information, see "Searching issues and pull requests."

Starting a review

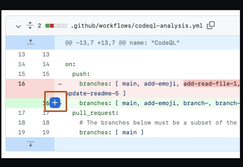
1. Under your repository name, click **Pull requests**.

2. In the list of pull requests, click the pull request you'd like to review.
3. On the pull request, click **Files changed**.

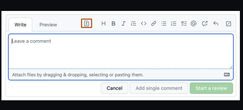
You can change the format of the diff view in this tab by clicking ⚙ and choosing the unified or split view. The choice you make will apply when you view the diff for other pull requests.

You can also choose to hide whitespace differences. The choice you make only applies to this pull request and will be remembered the next time you visit this page.

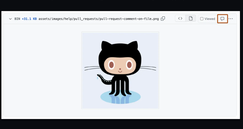
4. Optionally, filter the files to show only the files you want to review or use the file tree to navigate to a specific file. For more information, see "Filtering files in a pull request."
5. Hover over the line of code where you'd like to add a comment, and click the blue comment icon. To add a comment on multiple lines, click and drag to select the range of lines, then click the blue comment icon.



6. In the comment field, type your comment.
7. Optionally, to suggest a specific change to the line or lines, click , then edit the text within the suggestion block.

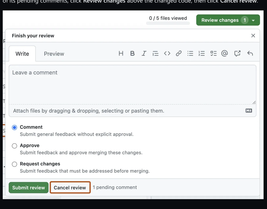


8. To comment directly on a file, to the right of the file, click and type your comment.



9. When you're done, click **Start a review**. If you have already started a review, you can click **Add review comment**.

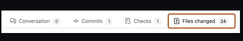
Before you submit your review, your line comments are *pending* and only visible to you. You can edit pending comments anytime before you submit your review. To cancel a pending review, including all of its pending comments, click **Review changes** above the changed code, then click **Cancel review**.



Reviewing dependency changes

If the pull request contains changes to dependencies you can use the dependency review for a manifest or lock file to see what has changed and check whether the changes introduce security vulnerabilities. For more information, see "Reviewing dependency changes in a pull request."

1. On the pull request, click **Files changed**.



2. On the right of the header for a manifest or lock file, display the dependency review by clicking the rich diff button.

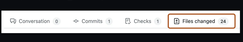
3. You may also want to review the source diff, because there could be changes to the manifest or lock file that don't change dependencies, or there could be dependencies that GitHub can't parse and which, as a result, don't appear in the dependency review.

   To return to the source diff view, click the button.

Marking a file as viewed

After you finish reviewing a file, you can mark the file as viewed, and the file will collapse. If the file changes after you view the file, it will be unmarked as viewed.

1. On the pull request, click **Files changed**.

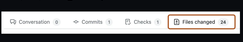


2. On the right of the header of the file you've finished reviewing, select **Viewed**.

Submitting your review

After you've finished reviewing all the files you want in the pull request, submit your review.

1. On the pull request, click **Files changed**.



2. Above the changed code, click **Review changes**.

3. Type a comment summarizing your feedback on the proposed changes.
4. Select the type of review you'd like to leave:
   - Select **Comment** to leave general feedback without explicitly approving the changes or requesting additional changes.
   - Select **Approve** to submit your feedback and approve merging the changes proposed in the pull request.
   - Select **Request changes** to submit feedback that must be addressed before the pull request can be merged.
5. Click **Submit review**.

Tips:

- If a collaborator with `admin`, `owner`, or `write` access to the repository submits a review requesting changes, the pull request cannot be merged until the same collaborator submits another review approving the changes in the pull request.
- Repository owners and administrators can merge a pull request even if it hasn't received an approving review, or if a reviewer who requested changes has left the organization or is unavailable.
- If both required reviews and stale review dismissal are enabled and a code-modifying commit is pushed to the branch of an approved pull request, the approval is dismissed. The pull request must be reviewed and approved again before it can be merged.
- When several open pull requests each have a head branch pointing to the same commit, you won't be able to merge them if one or both have a pending or rejected review.
- If your repository requires approving reviews from people with write or admin permissions, then any approvals from people with these permissions are denoted with a green check mark, and approvals from people without these permissions have a gray check mark. Approvals with a gray check mark do not affect whether the pull request can be merged.
- Pull request authors cannot approve their own pull requests.

Further reading

- "About protected branches"
- "Filtering and searching issues and pull requests"

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

Legal
© 2023 GitHub, Inc. Terms Privacy Status Pricing Expert services Blog