

Troubleshooting required status checks

In this article

Conflicts between head commit and test merge commit

Handling skipped but required checks

Required status checks from unexpected sources

You can check for common errors and resolve issues with required status checks.

Protected branches are available in public repositories with GitHub Free and GitHub Free for organizations, and in public and private repositories with GitHub Pro, GitHub Team, GitHub Enterprise Cloud, and GitHub Enterprise Server. For more information, see "[GitHub's plans](#)."

If you have a check and a status with the same name, and you select that name as a required status check, both the check and the status are required. For more information, see "[Checks](#)."

After you enable required status checks, your branch may need to be up-to-date with the base branch before merging. This ensures that your branch has been tested with the latest code from the base branch. If your branch is out of date, you'll need to merge the base branch into your branch. For more information, see "[About protected branches](#)."

Note: You can also bring your branch up to date with the base branch using Git rebase. For more information, see "[About Git rebase](#)."

You won't be able to push local changes to a protected branch until all required status checks pass. Instead, you'll receive an error message similar to the following.

```
remote: error: GH006: Protected branch update failed for refs/heads/main.
remote: error: Required status check "ci-build" is failing
```

Note: Pull requests that are up-to-date and pass required status checks can be merged locally and pushed to the protected branch. This can be done without status checks running on the merge commit itself.

Conflicts between head commit and test merge commit

Sometimes, the results of the status checks for the test merge commit and head commit will conflict. If the test merge commit has a status, the test merge commit must pass. Otherwise, the status of the head commit must pass before you can merge the branch.

If there is a conflict between the test merge commit and head commit, the checks for the test merge commit are shown in the pull request status checks box. This is indicated in the pull request status box by a line starting with `Showing checks for the merge commit`. For more information about test merge commits, see "[Pulls](#)."

Handling skipped but required checks

Warning: If a workflow is skipped due to [path filtering](#), [branch filtering](#) or a [commit message](#), then checks associated with that workflow will remain in a "Pending" state. A pull request that requires those checks to be successful will be blocked from merging.

For this reason you should not use path or branch filtering to skip workflow runs if the workflow is required. For more information, see "[Skipping workflow runs](#)" and "[Required workflows](#)."

If, however, a job within a workflow is skipped due to a conditional, it will report its status as "Success". For more information, see "[Using conditions to control job execution](#)."

Example

The following example shows a workflow that requires a "Successful" completion status for the `build` job, but the workflow will be skipped if the pull request does not change any files in the `scripts` directory.

```
name: ci
on:
  pull_request:
    paths:
      - 'scripts/**'
jobs:
  build:
    runs-on: ubuntu-latest
    strategy:
      matrix:
        node-version: [12.x, 14.x, 16.x]
    steps:
      - uses: actions/checkout@v4
      - name: Use Node.js ${ matrix.node-version }
        uses: actions/setup-node@v3
        with:
          node-version: ${ matrix.node-version }
          cache: 'npm'
      - run: npm ci
      - run: npm run build --if-present
      - run: npm test
```

Due to [path filtering](#), a pull request that only changes a file in the root of the repository will not trigger this workflow and is blocked from merging. On the pull request, you would see "Waiting for status to be reported."

It is recommended that you do not use path filtering (as shown in the previous example), or branch filtering, in a workflow that has been configured to be required. For more information, see "[Required workflows](#)."

Required status checks from unexpected sources

It's also possible for a protected branch to require a status check from a specific GitHub App. If you see a message similar to the following, then you should verify that the check listed in the merge box was set by the expected app.

Required status check "build" was not set by the expected GitHub App.

Legal

