

Reviewing dependency changes in a pull request

In this article

About dependency review

Reviewing dependencies in a pull request

If a pull request contains changes to dependencies, you can view a summary of what has changed and whether there are known vulnerabilities in any of the dependencies.

Dependency review is included in GitHub Enterprise Cloud for public repositories. To use dependency review in private repositories owned by organizations, you must have a license for GitHub Advanced Security. For more information, see "[About GitHub Advanced Security](#)."

About dependency review [↗](#)

Dependency review helps you understand dependency changes and the security impact of these changes at every pull request. It provides an easily understandable visualization of dependency changes with a rich diff on the "Files Changed" tab of a pull request.

Dependency review informs you of:

- Which dependencies were added, removed, or updated, along with the release dates.
- How many projects use these components.
- Vulnerability data for these dependencies.

Before you can use dependency review in a private repository, you must enable the dependency graph. For more information, see "[Exploring the dependencies of a repository](#)."


Dependency review allows you to "shift left". You can use the provided predictive information to catch vulnerable dependencies before they hit production. For more information, see "[About dependency review](#)."

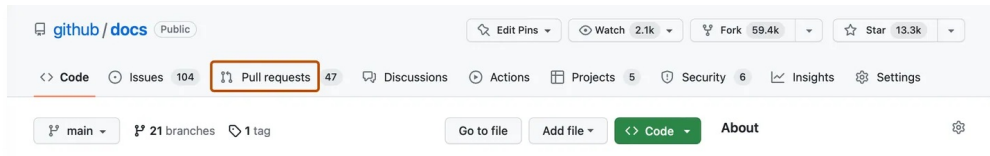
You can use the dependency review action to help enforce dependency reviews on pull requests in your repository. The dependency review action scans your pull requests for dependency changes and raises an error if any new dependencies have known vulnerabilities. The action is supported by an API endpoint that compares the dependencies between two revisions and reports any differences.

For more information about the action and the API endpoint, see the [dependency-review-action](#) documentation, and "[Dependency review](#)" in the API documentation.


You can configure the dependency review action to better suit your needs by specifying the type of dependency vulnerability you wish to catch. For more information, see "[Configuring dependency review](#)."

Reviewing dependencies in a pull request [↗](#)

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

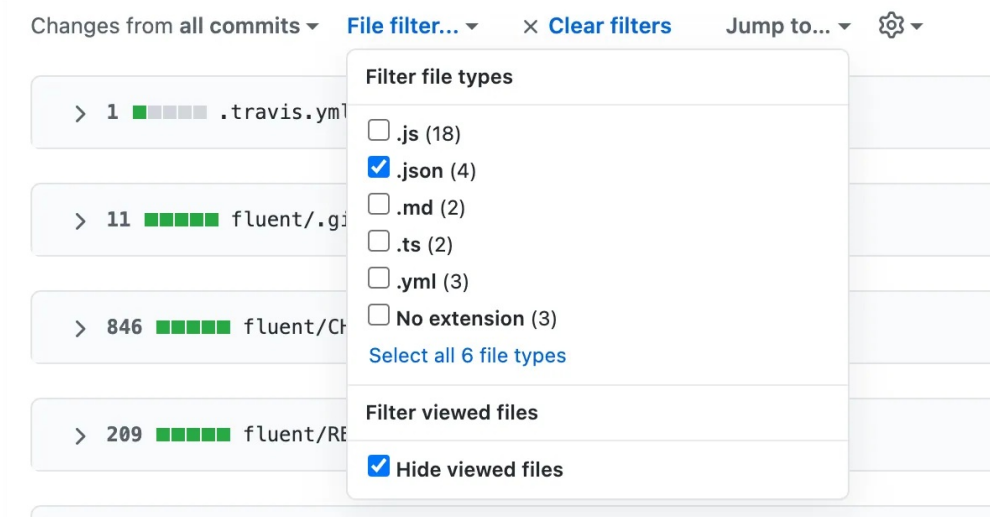


- 2 In the list of pull requests, click the pull request you'd like to review.

- 3 On the pull request, click  **Files changed**.




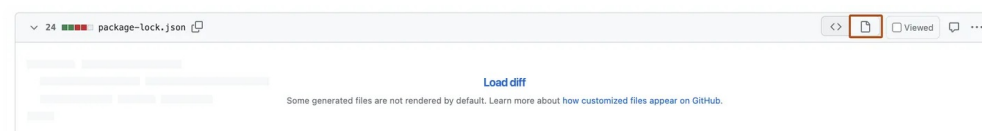
- 4 If the pull request contains many files, use the **File filter** drop-down menu to collapse all files that don't record dependencies. This will make it easier to focus your review on the dependency changes.



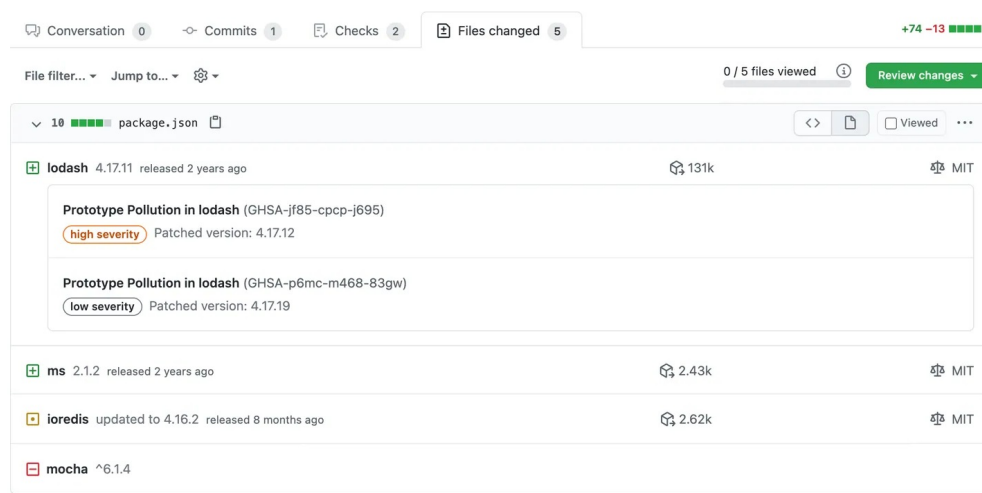
The dependency review provides a clearer view of what has changed in large lock files, where the source diff is not rendered by default.

Note: Dependency review rich diffs are not available for committed static JavaScript files like `jquery.js`.

- 5 On the right of the header for a manifest or lock file, display the dependency review by clicking .



- 6 Check the dependencies listed in the dependency review.



Any added or changed dependencies that have vulnerabilities are listed first, ordered by severity and then by dependency name. This means that the highest severity dependencies are always at the top of a dependency review. Other dependencies are listed alphabetically by dependency name.

The icon beside each dependency indicates whether the dependency has been added (+), updated (u), or removed (-) in this pull request.

Other information includes:

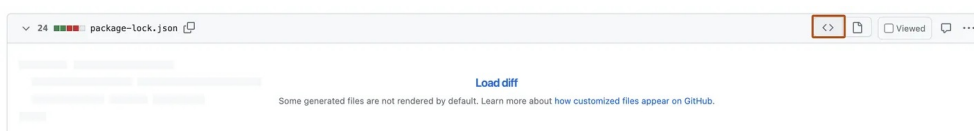
- The version, or version range, of the new, updated, or deleted dependency.
- For a specific version of a dependency:
 - The age of that release of the dependency.
 - The number of projects that are dependent on this software. This information is taken from the dependency graph. Checking the number of dependents can help you avoid accidentally adding the wrong dependency.
 - The license used by this dependency, if this information is available. This is useful if you want to avoid code with certain licenses being used in your project.

Where a dependency has a known vulnerability, the warning message includes:

- A brief description of the vulnerability.
- A Common Vulnerabilities and Exposures (CVE) or GitHub Security Advisories (GHSA) identification number. You can click this ID to find out more about the vulnerability.
- The severity of the vulnerability.
- The version of the dependency in which the vulnerability was fixed. If you are reviewing a pull request for someone, you might ask the contributor to update the dependency to the patched version, or a later release.

- 7 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.



Legal

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