

# Configuring the dependency graph

## In this article

About the dependency graph

About configuring the dependency graph

Further reading

You can allow users to identify their projects' dependencies by enabling the dependency graph.

## About the dependency graph

The dependency graph is a summary of the manifest and lock files stored in a repository and any dependencies that are submitted for the repository using the Dependency submission API (beta). For each repository, it shows:

- Dependencies, the ecosystems and packages it depends on
- Dependents, the repositories and packages that depend on it

For each dependency, you can see the license information and vulnerability severity. You can also search for a specific dependency using the search bar. Dependencies are sorted automatically by vulnerability severity.

For more information, see "[About the dependency graph](#)."

## About configuring the dependency graph

To generate a dependency graph, GitHub needs read-only access to the dependency manifest and lock files for a repository. The dependency graph is automatically generated for all public repositories and you can choose to enable it for private repositories. For more information on viewing the dependency graph, see "[Exploring the dependencies of a repository](#)."


Additionally, you can use the Dependency submission API (beta) to submit dependencies from the package manager or ecosystem of your choice, even if the ecosystem is not supported by dependency graph for manifest or lock file analysis. Dependencies submitted to a project using the Dependency submission API (beta) will show which detector was used for their submission and when they were submitted. For more information on the Dependency submission API, see "[Using the Dependency submission API](#)."

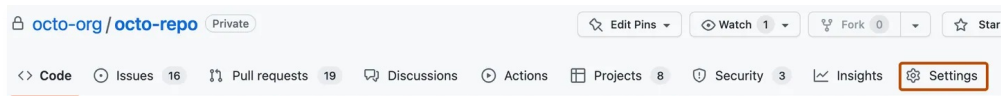
## Enabling and disabling the dependency graph for a private repository


Repository administrators can enable or disable the dependency graph for private repositories.

You can enable or disable the dependency graph for all repositories owned by your user account. For more information, see "[Managing security and analysis settings for your personal account](#)".

You can also enable the dependency graph for multiple repositories in an organization at the same time. For more information, see "[Securing your organization](#)."

- 1 On GitHub.com, navigate to the main page of the repository.
- 2 Under your repository name, click  **Settings**. If you cannot see the "Settings" tab, select the ... dropdown menu, then click **Settings**.



- 3 In the "Security" section of the sidebar, click  **Code security and analysis**.
- 4 Read the message about granting GitHub read-only access to the repository data to enable the dependency graph, then next to "Dependency Graph", click **Enable**.

## Code security and analysis

Security and analysis features help keep your repository secure and updated. By enabling these features, you're granting us permission to perform read-only analysis on your repository.

### Dependency graph

Understand your dependencies.

**Enable**

You can disable the dependency graph at any time by clicking **Disable** next to "Dependency Graph" on the settings page for "Code security and analysis."

When the dependency graph is first enabled, any manifest and lock files for supported ecosystems are parsed immediately. The graph is usually populated within minutes but this may take longer for repositories with many dependencies. Once enabled, the graph is automatically updated with every push to the repository and every push to other repositories in the graph.

## Further reading

- "[Viewing and updating Dependabot alerts](#)"
- "[Troubleshooting the detection of vulnerable dependencies](#)"

## Legal

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