Managing pull requests for dependency updates

In this article

You manage pull requests raised by Dependabot in much the same way as other pull requests, but there are some extra options.

> **Note:** Your site administrator must set up Dependabot updates for your GitHub Enterprise Server instance before you can use this feature. For more information, see "Enabling Dependabot for your enterprise."
>
> You may not be able to enable or disable Dependabot updates if an enterprise owner has set a policy at the enterprise level. For more information, see "Enforcing policies for code security and analysis for your enterprise."

About Dependabot pull requests 🔗

Dependabot raises pull requests to update dependencies. Depending on how your repository is configured, Dependabot may raise pull requests for version updates and/or for security updates. You manage these pull requests in the same way as any other pull request, but there are also some extra commands available. For information about enabling Dependabot dependency updates, see "Configuring Dependabot security updates" and "Configuring Dependabot version updates."

When Dependabot raises a pull request, you're notified by your chosen method for the repository. Each pull request contains detailed information about the proposed change, taken from the package manager. These pull requests follow the normal checks and tests defined in your repository.

If you have many dependencies to manage, you may want to customize the configuration for each package manager so that pull requests have specific reviewers, assignees, and labels. For more information, see "Customizing dependency updates."

> **Note**: If you don't interact with Dependabot pull requests for a repository during a 90-day time period, Dependabot considers your repository as inactive, and will automatically pause Dependabot updates. For more information about inactivity criteria, see "About Dependabot version updates" and "About Dependabot security updates."

Viewing Dependabot pull requests 🔗

1. On your GitHub Enterprise Server instance, navigate to the main page of the repository.

2. Under your repository name, click ⑂ **Pull requests**.
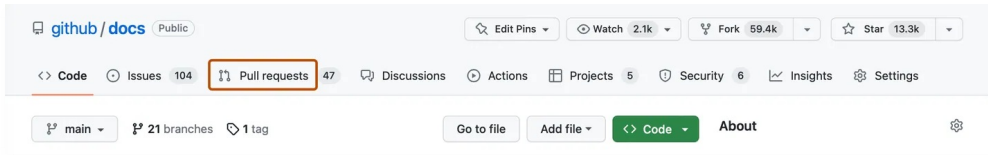
3. Any pull requests for security or version updates are easy to identify.

   - The author is dependabot, the bot account used by Dependabot.
   - By default, they have the `dependencies` label.

Changing the rebase strategy for Dependabot pull requests 🔗

By default, Dependabot automatically rebases pull requests to resolve any conflicts. If a pull request has not been merged for 30 days, Dependabot will stop rebasing the pull request. You can still manually rebase and merge the pull request. If you'd prefer to handle merge conflicts manually, you can disable this using the `rebase-strategy` option. For details, see "[Configuration options for the dependabot.yml file](#)."

Allowing Dependabot to rebase and force push over extra commits 🔗

By default, Dependabot will stop rebasing a pull request once extra commits have been pushed to it. To allow Dependabot to force push over commits added to its branches, include any of the following strings: `[dependabot skip]` , `[skip dependabot]` , `[dependabot-skip]` , or `[skip-dependabot]` , in either lower or uppercase, to the commit message.

Managing Dependabot pull requests with comment commands 🔗

Dependabot responds to simple commands in comments. Each pull request contains details of the commands you can use to process the pull request (for example: to merge, squash, reopen, close, or rebase the pull request) under the "Dependabot commands and options" section. The aim is to make it as easy as possible for you to triage these automatically generated pull requests.

You can use any of the following commands on a Dependabot pull request.

- `@dependabot cancel merge` cancels a previously requested merge.
- `@dependabot close` closes the pull request and prevents Dependabot from recreating that pull request. You can achieve the same result by closing the pull request manually.
- `@dependabot ignore this dependency` closes the pull request and prevents Dependabot from creating any more pull requests for this dependency (unless you reopen the pull request or upgrade to the suggested version of the dependency yourself).
- `@dependabot ignore this major version` closes the pull request and prevents

Dependabot from creating any more pull requests for this major version (unless you reopen the pull request or upgrade to this major version yourself).

- `@dependabot ignore this minor version` closes the pull request and prevents Dependabot from creating any more pull requests for this minor version (unless you reopen the pull request or upgrade to this minor version yourself).
- `@dependabot merge` merges the pull request once your CI tests have passed.
- `@dependabot rebase` rebases the pull request.
- `@dependabot recreate` recreates the pull request, overwriting any edits that have been made to the pull request.
- `@dependabot reopen` reopens the pull request if the pull request is closed.
- `@dependabot squash and merge` squashes and merges the pull request once your CI tests have passed.

Dependabot will react with a "thumbs up" emoji to acknowledge the command, and may respond with a comment on the pull request. While Dependabot usually responds quickly, some commands may take several minutes to complete if Dependabot is busy processing other updates or commands.

If you run any of the commands for ignoring dependencies or versions, Dependabot stores the preferences for the repository centrally. While this is a quick solution, for repositories with more than one contributor it is better to explicitly define the dependencies and versions to ignore in the configuration file. This makes it easy for all contributors to see why a particular dependency isn't being updated automatically.

For more information, see "[Configuration options for the dependabot.yml file](#)."