Managing notifications from your inbox

In this article

Use your inbox to quickly triage and sync your notifications across email and mobile.

About your inbox

To use the notifications inbox on GitHub and GitHub Mobile, you must enable web and mobile notifications in your notification settings. For more information, see "[Configuring notifications](#)."

To access your notifications inbox, in the upper-right corner of any page, click 🔔.

Your inbox shows all of the notifications that you haven't unsubscribed to or marked as **Done.** You can customize your inbox to best suit your workflow using filters, viewing all or just unread notifications, and grouping your notifications to get a quick overview.

By default, your inbox will show read and unread notifications. To only see unread notifications, click **Unread** or use the `is:unread` query.

Triaging options

You have several options for triaging notifications from your inbox.

Triaging option	Description
Save	Saves your notification for later review. To save a notification, to the right of the notification, click 🔖.
	Saved notifications are kept indefinitely and can be viewed by clicking **Saved** in the sidebar or with the `is:saved` query. If your saved notification is older than 5 months and becomes unsaved, the notification will disappear from your inbox within a day.
Done	Marks a notification as completed and removes the notification from your inbox. You can see all completed notifications by clicking **Done** in the

sidebar or with the `is:done` query. Notifications marked as **Done** are saved for 5 months.
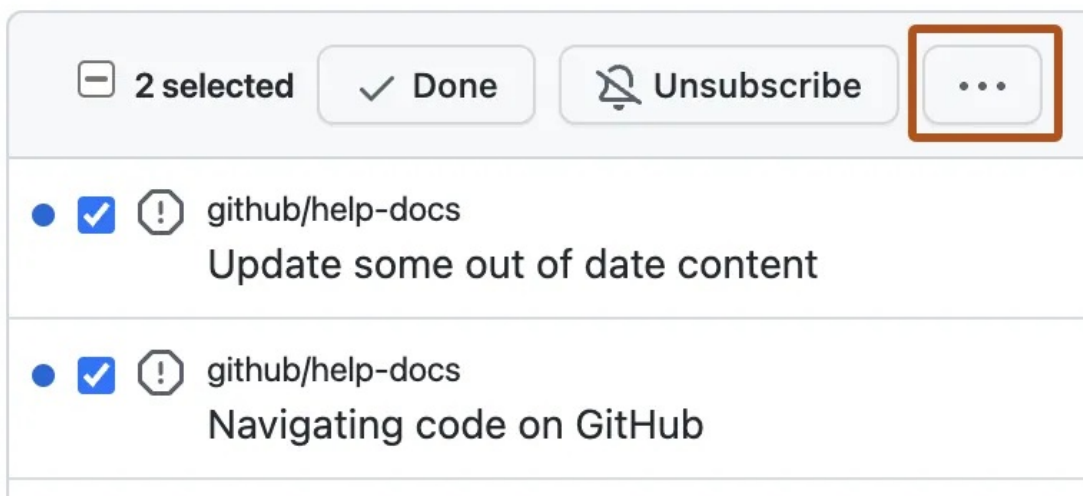
Unsubscribe	Automatically removes the notification from your inbox and unsubscribes you from the conversation until you are @mentioned, a team you're on is @mentioned, or you're requested for review.
Read	Marks a notification as read. To only view read notifications in your inbox, use the `is:read` query. This query doesn't include notifications marked as **Done**.
Unread	Marks notification as unread. To only view unread notifications in your inbox, use the `is:unread` query.

To see the available keyboard shortcuts, see "Keyboard shortcuts."

Before choosing a triage option, you can preview your notification's details first and investigate. For more information, see "Triaging a single notification."

Triaging multiple notifications at the same time 🔗

To triage multiple notifications at once, select the relevant notifications and use the ⋯ drop-down to choose a triage option.
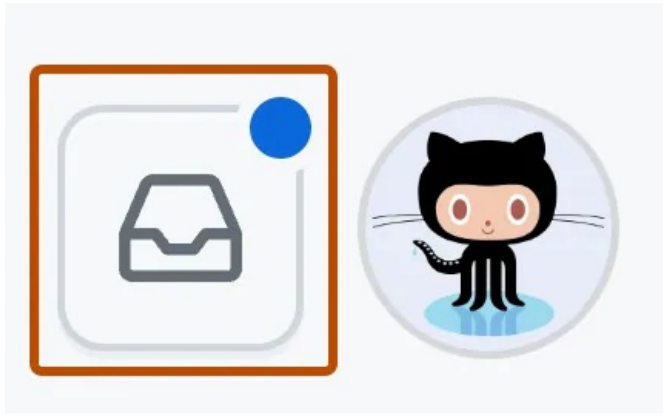

Default notification filters 🔗

By default, your inbox has filters for when you are assigned, participating in a thread, requested to review a pull request, or when your username is @mentioned directly or a team you're a member of is @mentioned.

Customizing your inbox with custom filters 🔗
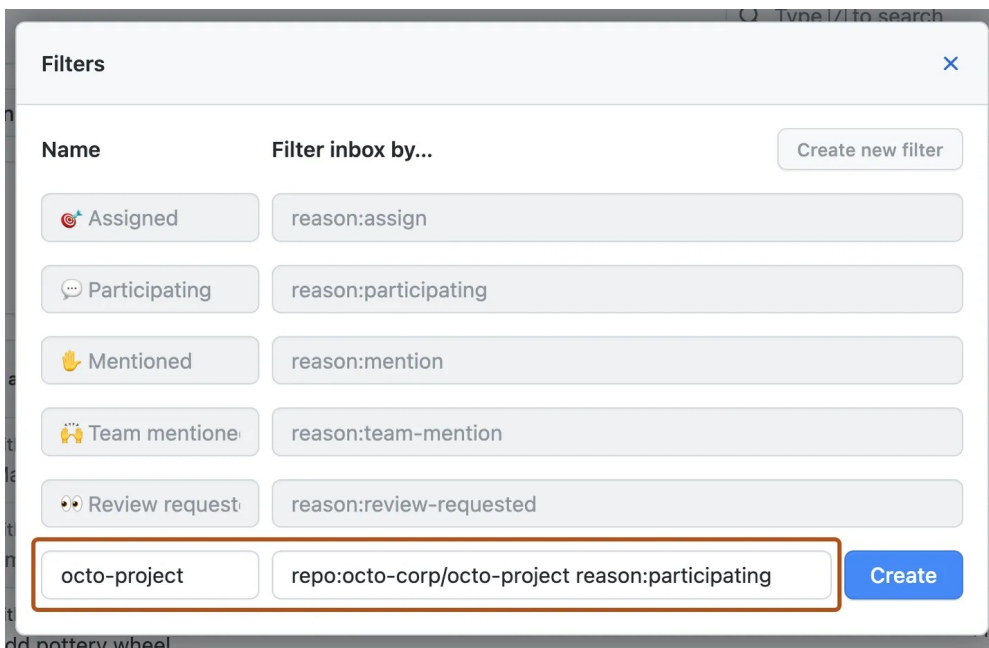
You can add up to 15 of your own custom filters.

1. In the upper-right corner of any page, click ✉.

2 To open the filter settings, in the left sidebar, next to "Filters", click ⚙.

> **Tip:** You can quickly preview a filter's inbox results by creating a query in your inbox view and clicking **Save**, which opens the custom filter settings.

3 Add a name for your filter and a filter query. For example, to only see notifications for a specific repository, you can create a filter using the query `repo:octocat/open-source-project-name reason:participating`. You can also add emojis with a native emoji keyboard. For a list of supported search queries, see "[Supported queries for custom filters](#)."

4 Click **Create**.

Custom filter limitations 🔗

Custom filters do not currently support:

- Full text search in your inbox, including searching for pull request or issue titles.
- Distinguishing between the `is:issue`, `is:pr`, and `is:pull-request` query filters. These queries will return both issues and pull requests.
- Creating more than 15 custom filters.
- Changing the default filters or their order.
- Search [exclusion](#) using `NOT` or `-QUALIFIER`.

Supported queries for custom filters 🔗

These are the types of filters that you can use:

- Filter by repository with `repo:`
- Filter by discussion type with `is:`
- Filter by notification reason with `reason:`
- Filter by notification author with `author:`
- Filter by organization with `org:`

Supported `repo:` queries 🔗

To add a `repo:` filter, you must include the owner of the repository in the query: `repo:owner/repository`. An owner is the organization or the user who owns the GitHub asset that triggers the notification. For example, `repo:octo-org/octo-repo` will show notifications triggered in the octo-repo repository within the octo-org organization.

Supported `is:` queries 🔗

To filter notifications for specific activity on GitHub.com, you can use the `is` query. For example, to only see repository invitation updates, use `is:repository-invitation`, and to only see Dependabot alerts, use `is:repository-vulnerability-alert`.

- `is:check-suite`
- `is:commit`
- `is:gist`
- `is:issue-or-pull-request`
- `is:release`
- `is:repository-invitation`
- `is:repository-vulnerability-alert`
- `is:repository-advisory`
- `is:discussion`

For information about reducing noise from notifications for Dependabot alerts, see "[Configuring notifications for Dependabot alerts]."

You can also use the `is:` query to describe how the notification was triaged.

- `is:saved`
- `is:done`
- `is:unread`
- `is:read`

Supported `reason:` queries 🔗

To filter notifications by why you've received an update, you can use the `reason:` query. For example, to see notifications when you (or a team you're on) is requested to review a pull request, use `reason:review-requested`. For more information, see "[About notifications]."

Query	Description
`reason:assign`	When there's an update on an issue or pull request you've been assigned to.
`reason:author`	When you opened a pull request or issue and there has been an update or new comment.

`reason:comment`	When you commented on an issue or pull request.
`reason:participating`	When you have commented on an issue or pull request or you have been @mentioned.
`reason:invitation`	When you're invited to a team, organization, or repository.
`reason:manual`	When you click **Subscribe** on an issue or pull request you weren't already subscribed to.
`reason:mention`	You were directly @mentioned.
`reason:review-requested`	You or a team you're on have been requested to review a pull request.
`reason:security-alert`	When a security alert is issued for a repository.
`reason:state-change`	When the state of a pull request or issue is changed. For example, an issue is closed or a pull request is merged.
`reason:team-mention`	When a team you're a member of is @mentioned.
`reason:ci-activity`	When a repository has a CI update, such as a new workflow run status.

Supported `author:` queries 🔗

To filter notifications by user, you can use the `author:` query. An author is the original author of the thread (issue, pull request, gist, discussions, and so on) for which you are being notified. For example, to see notifications for threads created by the Octocat user, use `author:octocat` .

Supported `org:` queries 🔗

To filter notifications by organization, you can use the `org` query. The organization you need to specify in the query is the organization of the repository for which you are being notified on GitHub. This query is useful if you belong to several organizations, and want to see notifications for a specific organization.

For example, to see notifications from the octo-org organization, use `org:octo-org` .

Dependabot custom filters 🔗

If you use Dependabot to keep your dependencies up-to-date, you can use and save these custom filters:

- `is:repository_vulnerability_alert` to show notifications for Dependabot alerts.
- `reason:security_alert` to show notifications for Dependabot alerts and security update pull requests.
- `author:app/dependabot` to show notifications generated by Dependabot. This includes Dependabot alerts, security update pull requests, and version update pull requests.

For more information about Dependabot, see "[About Dependabot alerts](#)."