Reviewing the audit log for your organization

In this article

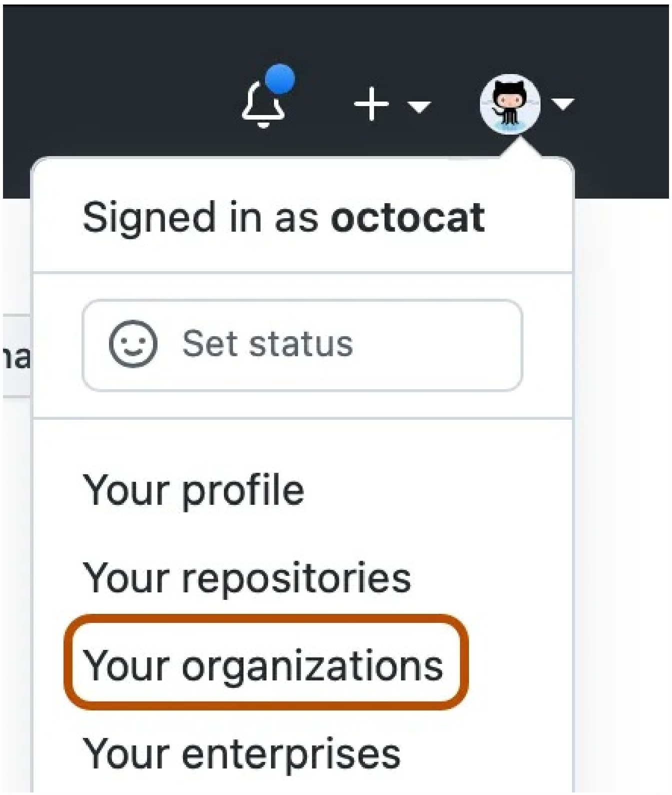
The audit log allows organization admins to quickly review the actions performed by members of your organization. It includes details such as who performed the action, what the action was, and when it was performed.

Accessing the audit log

The audit log lists events triggered by activities that affect your organization within the current month and previous six months. Only owners can access an organization's audit log.

By default, only events from the past three months are displayed. To view older events, you must specify a date range with the `created` parameter. For more information, see "Understanding the search syntax."

1. In the top right corner of GitHub Enterprise Server, click your profile photo, then click **Your organizations**.



2. Next to the organization, click **Settings**.

3. In the "Archives" section of the sidebar, click **Logs**, then click **Audit log**.

Searching the audit log

The name for each audit log entry is composed of a category of events, followed by an operation type. For example, the `repo.create` entry refers to the `create` operation on the `repo` category.

Each audit log entry shows applicable information about an event, such as:

- The enterprise or organization an action was performed in
- The user (actor) who performed the action
- The user affected by the action
- Which repository an action was performed in
- The action that was performed
- Which country the action took place in
- The date and time the action occurred
- For actions outside of the web UI, how the user (actor) authenticated

Note that you cannot search for entries using text. You can, however, construct search queries using a variety of filters. Many operators used when querying the log, such as `-`, `>`, or `<`, match the same format as searching across GitHub Enterprise Server. For more information, see "[About searching on GitHub](#)."

Search based on operation

Use the `operation` qualifier to limit actions to specific types of operations. For example:

- `operation:access` finds all events where a resource was accessed.
- `operation:authentication` finds all events where an authentication event was performed.
- `operation:create` finds all events where a resource was created.
- `operation:modify` finds all events where an existing resource was modified.
- `operation:remove` finds all events where an existing resource was removed.
- `operation:restore` finds all events where an existing resource was restored.
- `operation:transfer` finds all events where an existing resource was transferred.

Search based on repository

Use the `repo` qualifier to limit actions to a specific repository. For example:

- `repo:"my-org/our-repo"` finds all events that occurred for the `our-repo` repository in the `my-org` organization.
- `repo:"my-org/our-repo" repo:"my-org/another-repo"` finds all events that occurred for both the `our-repo` and `another-repo` repositories in the `my-org` organization.
- `-repo:"my-org/not-this-repo"` excludes all events that occurred for the `not-this-repo` repository in the `my-org` organization.

Note that you must include the account name within the `repo` qualifier and put it in quotes or escape the `/` with a `\`; searching for just `repo:our-repo` or `repo:my-org/our-repo` will not work.

Search based on the user

The `actor` qualifier can scope events based on who performed the action. For example:

- `actor:octocat` finds all events performed by `octocat`.
- `actor:octocat actor:hubot` finds all events performed by `octocat` or `hubot`.
- `-actor:hubot` excludes all events performed by `hubot`.

Note that you can only use a GitHub Enterprise Server username, not an individual's real name.

Search based on the action performed

To search for specific events, use the `action` qualifier in your query. Actions listed in the audit log are grouped in different categories. For the full list of events in each category, see "Audit log events for your organization."

Category name	Description
`auto_approve_personal_access_token_requests`	Contains activities related to your organization's approval policy for fine-grained personal access tokens. For more information, see "Setting a personal access token policy for your organization."
`copilot`	Contains all activities related to your GitHub Copilot for Business subscription.
`dependabot_alerts`	Contains organization-level configuration activities for Dependabot alerts in existing repositories. For more information, see "About Dependabot alerts."
`dependabot_alerts_new_repos`	Contains organization-level configuration activities for Dependabot alerts in new repositories created in the organization.
`dependabot_security_updates`	Contains organization-level configuration activities for Dependabot security updates in existing repositories. For more information, see "Configuring Dependabot security updates."
`dependabot_security_updates_new_repos`	Contains organization-level configuration activities for Dependabot security updates for new repositories created in the organization.
`discussion_post`	Contains all activities related to discussions posted to a team page.
`discussion_post_reply`	Contains all activities related to replies to discussions posted to a team page.
`enterprise`	Contains activities related to enterprise settings.
`hook`	Contains all activities related to webhooks.
`integration_installation`	Contains activities related to integrations installed in an account.
`integration_installation_request`	Contains all activities related to organization member requests for owners to approve integrations for use in the organization.

`issue`	Contains activities related to deleting an issue.
`members_can_create_pages`	Contains all activities related to managing the publication of GitHub Pages sites for repositories in the organization. For more information, see "Managing the publication of GitHub Pages sites for your organization."
`org`	Contains activities related to organization membership.
`org_secret_scanning_custom_pattern`	Contains organization-level activities related to secret scanning custom patterns. For more information, see "Defining custom patterns for secret scanning."
`organization_default_label`	Contains all activities related to default labels for repositories in your organization.
`oauth_application`	Contains all activities related to OAuth apps.
`packages`	Contains all activities related to GitHub Packages.
`personal_access_token`	Contains activities related to fine-grained personal access tokens in your organization. For more information, see "Managing your personal access tokens."
`profile_picture`	Contains all activities related to your organization's profile picture.
`project`	Contains all activities related to project boards.
`protected_branch`	Contains all activities related to protected branches.
`repo`	Contains activities related to the repositories owned by your organization.
`repository_secret_scanning`	Contains repository-level activities related to secret scanning. For more information, see "About secret scanning."
`repository_secret_scanning_custom_pattern`	Contains repository-level activities related to secret scanning custom patterns. For more information, see "Defining custom patterns for secret scanning."
`repository_secret_scanning_custom_pattern_push_protection`	Contains repository-level activities related to push protection of a custom pattern for secret scanning. For more information, see "Defining custom patterns for secret scanning."
`repository_secret_scanning_push_protection`	Contains repository-level activities related to secret scanning push protection. For more information, see "Push protection for repositories and organizations."
`repository_vulnerability_alert`	Contains all activities related to Dependabot alerts.
`role`	Contains all activities related to custom repository roles.

`secret_scanning`	Contains organization-level configuration activities for secret scanning in existing repositories. For more information, see "About secret scanning."
`secret_scanning_new_repos`	Contains organization-level configuration activities for secret scanning for new repositories created in the organization.
`team`	Contains all activities related to teams in your organization.
`team_discussions`	Contains activities related to managing team discussions for an organization.
`workflows`	Contains activities related to GitHub Actions workflows.

You can search for specific sets of actions using these terms. For example:

- `action:team` finds all events grouped within the team category.
- `-action:hook` excludes all events in the webhook category.

Each category has a set of associated actions that you can filter on. For example:

- `action:team.create` finds all events where a team was created.
- `-action:hook.events_changed` excludes all events where the events on a webhook have been altered.

Search based on time of action

Use the `created` qualifier to filter events in the audit log based on when they occurred. Date formatting must follow the ISO8601 standard, which is `YYYY-MM-DD` (year-month-day). You can also add optional time information `THH:MM:SS+00:00` after the date, to search by the hour, minute, and second. That's `T`, followed by `HH:MM:SS` (hour-minutes-seconds), and a UTC offset (`+00:00`).

When you search for a date, you can use greater than, less than, and range qualifiers to further filter results. For more information, see "Understanding the search syntax."

For example:

- `created:2014-07-08` finds all events that occurred on July 8th, 2014.
- `created:>=2014-07-08` finds all events that occurred on or after July 8th, 2014.
- `created:<=2014-07-08` finds all events that occurred on or before July 8th, 2014.
- `created:2014-07-01..2014-07-31` finds all events that occurred in the month of July 2014.

Note: The audit log contains data for the current month and every day of the previous six months.

Search based on location

Using the qualifier `country`, you can filter events in the audit log based on the originating country. You can use a country's two-letter short code or its full name. Keep in mind that countries with spaces in their name will need to be wrapped in quotation marks. For example:

- `country:de` finds all events that occurred in Germany.

- `country:Mexico` finds all events that occurred in Mexico.
- `country:"United States"` all finds events that occurred in the United States.

Using the audit log API

You can interact with the audit log using the GraphQL API. You can use the `read:audit_log` scope to access the audit log via the APIs.

To ensure your intellectual property is secure, and you maintain compliance for your organization, you can use the audit log GraphQL API to keep copies of your audit log data and monitor:

- Access to your organization or repository settings
- Changes in permissions
- Added or removed users in an organization, repository, or team
- Users being promoted to admin
- Changes to permissions of a GitHub App

The GraphQL response can include data for up to 90 to 120 days.

For example, you can make a GraphQL request to see all the new organization members added to your organization. For more information, see the "Interfaces."

Further reading

- "Keeping your organization secure"