**This version of GitHub Enterprise was discontinued on 2023-03-15.** No patch releases will be made, even for critical security issues. For better performance, improved security, and new features, upgrade to the latest version of GitHub Enterprise. For help with the upgrade, contact GitHub Enterprise support.

# Assigning permissions to jobs

**In this article**

Modify the default permissions granted to `GITHUB_TOKEN`.

**Note:** GitHub-hosted runners are not currently supported on GitHub Enterprise Server. You can see more information about planned future support on the GitHub public roadmap.

## Overview

You can use `permissions` to modify the default permissions granted to the `GITHUB_TOKEN`, adding or removing access as required, so that you only allow the minimum required access. For more information, see "Automatic token authentication."

You can use `permissions` either as a top-level key, to apply to all jobs in the workflow, or within specific jobs. When you add the `permissions` key within a specific job, all actions and run commands within that job that use the `GITHUB_TOKEN` gain the access rights you specify. For more information, see `jobs.<job_id>.permissions`.

Available scopes and access values:

```
permissions:
  actions: read|write|none
  checks: read|write|none
  contents: read|write|none
  deployments: read|write|none
  issues: read|write|none
  discussions: read|write|none
  packages: read|write|none
  pages: read|write|none
  pull-requests: read|write|none
  repository-projects: read|write|none
  security-events: read|write|none
  statuses: read|write|none
```

If you specify the access for any of these scopes, all of those that are not specified are set to `none`.

You can use the following syntax to define read or write access for all of the available scopes:

```
permissions: read-all|write-all
```

You can use the following syntax to disable permissions for all of the available scopes:

```
permissions: {}
```

You can use the `permissions` key to add and remove read permissions for forked repositories, but typically you can't grant write access. The exception to this behavior is where an admin user has selected the **Send write tokens to workflows from pull requests** option in the GitHub Actions settings. For more information, see "Managing GitHub Actions settings for a repository."

### Example: Assigning permissions to GITHUB_TOKEN

This example shows permissions being set for the `GITHUB_TOKEN` that will apply to all jobs in the workflow. All permissions are granted read access.

```
name: "My workflow"

on: [ push ]

permissions: read-all

jobs:
  ...
```

## Assigning permissions to a specific job

For a specific job, you can use `jobs.<job_id>.permissions` to modify the default permissions granted to the `GITHUB_TOKEN`, adding or removing access as required, so that you only allow the minimum required access. For more information, see "Automatic token authentication."

By specifying the permission within a job definition, you can configure a different set of permissions for the `GITHUB_TOKEN` for each job, if required. Alternatively, you can specify the permissions for all jobs in the workflow. For information on defining permissions at the workflow level, see `permissions`.

Available scopes and access values:

```
permissions:
  actions: read|write|none
  checks: read|write|none
  contents: read|write|none
  deployments: read|write|none
  issues: read|write|none
  discussions: read|write|none
  packages: read|write|none
  pages: read|write|none
  pull-requests: read|write|none
  repository-projects: read|write|none
  security-events: read|write|none
  statuses: read|write|none
```

If you specify the access for any of these scopes, all of those that are not specified are set to `none`.

You can use the following syntax to define read or write access for all of the available scopes:

```
permissions: read-all|write-all
```

You can use the following syntax to disable permissions for all of the available scopes:

```
permissions: {}
```

You can use the `permissions` key to add and remove read permissions for forked repositories, but typically you can't grant write access. The exception to this behavior is where an admin user has selected the **Send write tokens to workflows from pull requests** option in the GitHub Actions settings. For more information, see "Managing GitHub Actions settings for a repository."

## Example: Setting permissions for a specific job

This example shows permissions being set for the `GITHUB_TOKEN` that will only apply to the job named `stale`. Write access is granted for the `issues` and `pull-requests` scopes. All other scopes will have no access.

```
jobs:
  stale:
    runs-on: ubuntu-latest

    permissions:
      issues: write
      pull-requests: write

    steps:
      - uses: actions/stale@v4
```