GitHub Docs

GitHub Packages / Learn GitHub Packages / Access control & visibility

Configuring a package's access control and visibility

In this article

Choose who has read, write, or admin access to your package and the visibility of your packages on GitHub.

GitHub Packages is available with GitHub Free, GitHub Pro, GitHub Free for organizations, GitHub Team, GitHub Enterprise Cloud, GitHub Enterprise Server 3.0 or higher, and GitHub AE.

GitHub Packages is not available for private repositories owned by accounts using legacy per-repository plans. Also, accounts using legacy per-repository plans cannot access registries that support granular permissions, because these accounts are billed by repository. For the list of registries that support granular permissions, see "About permissions for GitHub Packages." For more information, see "GitHub’s plans."

A package can inherit its visibility and access permissions from a repository, or, for registries that support granular permissions, you can set the visibility and permissions of the package separately from a repository.

For the list of registries that support granular permissions, and for more information about permissions for packages, packages-related scopes for PATs, or managing permissions for your GitHub Actions workflows, see "About permissions for GitHub Packages."

About inheritance of access permissions

In registries that support granular permissions, packages are scoped to a personal account or organization. In these registries, you can publish a package without linking the package to a repository, then determine who can access the package by setting access permissions and visibility in the package's settings.

By default, if you publish a package that is linked to a repository, the package automatically inherits the access permissions (but not the visibility) of the linked repository. For example, a user who has read access to the linked repository will also have read access to the package. When a package automatically inherits access permissions, GitHub Actions workflows in the linked repository also automatically get access to the package.

A package only inherits the access permissions of a linked repository automatically if you link the repository to the package before you publish the package, such as by adding the `org.opencontainers.image.source` Docker label to a container image. If you connect a published package to a repository from the package's settings page, the package will retain its existing access permissions, and will not inherit the access permissions of the repository unless you explicitly select this option. Additionally, organizations can disable automatic inheritance of access permissions for all new packages scoped to their organization. For more information, see "Disabling automatic inheritance of access permissions in an organization" below.

When a package inherits permissions from a repository, to grant or remove access to your package, you must configure the permissions settings of the linked repository. If you want to set a package's access settings separately from the repository linked to the package, you must remove the inherited permissions from the package. For more information, see "Selecting whether a package inherits permissions from a repository" below.

If you publish a package in a registry that only supports repository-scoped permissions, the package is always linked to a repository, and always inherits the permissions of the linked repository.

About setting visibility and access permissions for packages

If a package belongs to a registry that supports granular permissions, anyone with admin permissions to the package can set the package to private or public, and can grant access permissions for the package that are separate from the permissions set at the organization and repository levels. For the list of registries that support granular permissions, see "About permissions for GitHub Packages."

In most registries, to pull a package, you must authenticate with a personal access token or `GITHUB_TOKEN`, regardless of whether the package is public or private. However, in the Container registry, public packages allow anonymous access and can be pulled without authentication or signing in via the CLI.

> **Note:** If you publish a package that is linked to a repository, the package inherits its permissions from the linked repository by default. To access the package's granular permissions settings, you must remove the package's inherited permissions. If you're the owner of an organization, you can disable the automatic inheritance of permissions for all new packages scoped to your organization. For more information, see "Configuring a package's access control and visibility" and "Configuring a package's access control and visibility."

When you publish a package, you automatically get admin permissions to the package. If you publish a package to an organization, anyone with the `owner` role in the organization also gets admin permissions to the package.

For packages scoped to a personal account, you can give any person an access role. For packages scoped to an organization, you can give any person or team in the organization an access role.

If you are using a GitHub Actions workflow to manage your packages, you can grant an access role to the repository the workflow is stored in by using the **Add Repository** button under "Manage Actions access" in the package's settings. For more information, see "Configuring a package's access control and visibility."

Permission	Access description

Read	Can download package. Can read package metadata.
Write	Can upload and download this package. Can read and write package metadata.
Admin	Can upload, download, delete, and manage this package. Can read and write package metadata. Can grant package permissions.

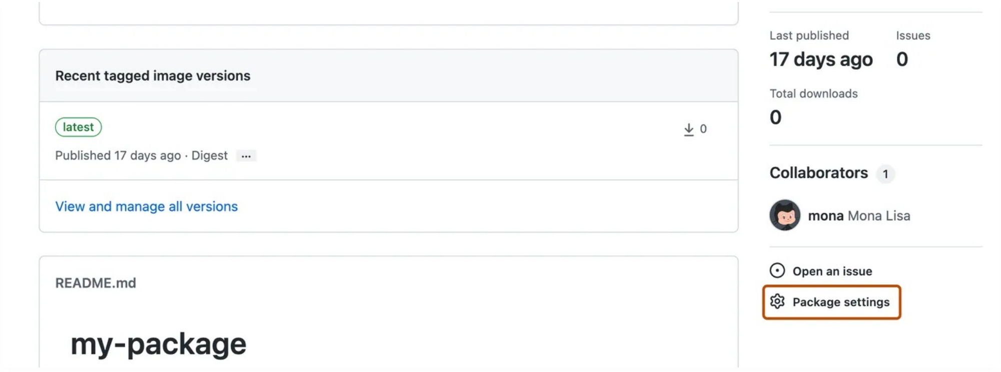
> **Note:** The ability for GitHub Actions workflows to delete and restore packages using the REST API is currently in public beta and subject to change.

Configuring access to packages for your personal account

If you have admin permissions to a package that's scoped to a personal account, you can assign read, write, or admin roles to other users. For more information about these permission roles, see "Visibility and access permissions for packages."

If your package is private or internal and scoped to an organization, then you can only give access to other organization members or teams.

1. Search for and then click the name of the package that you want to manage.
2. On your package's landing page, on the right-hand side, click ⚙ **Package settings**.



3. Under "Manage access" or "Inherited access", click **Invite teams or people** and enter the name, username, or email of the person you want to give access. Teams cannot be given access to a package that is scoped to a personal account.
4. Next to the username or team name, use the **Role** drop-down menu to select a desired permission level.

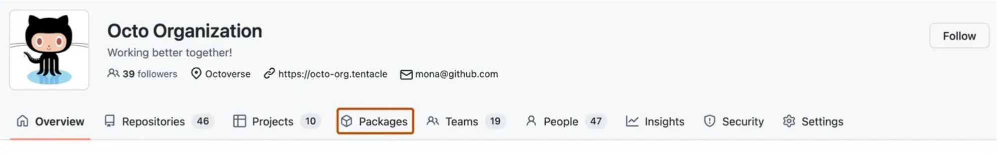
The selected users will automatically be given access and don't need to accept an invitation first.

Configuring access to packages for an organization

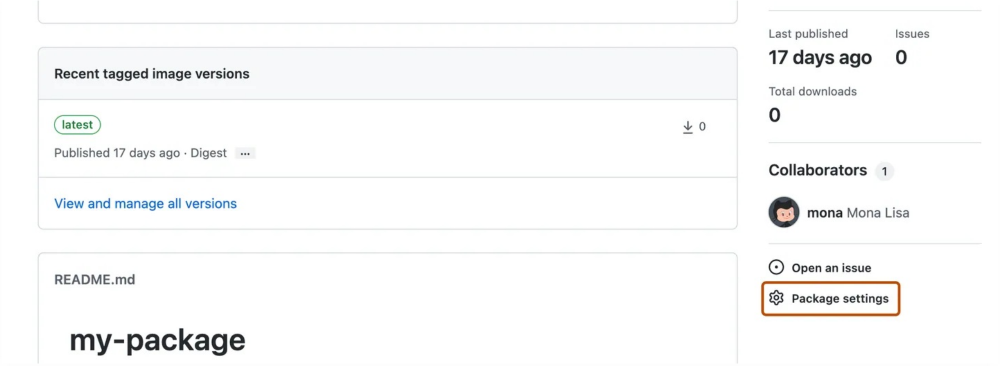
If you have admin permissions to a package that is scoped to an organization, you can assign read, write, or admin roles to other users and teams. For more information about these permission roles, see "Visibility and access permissions for packages."

If your package is private or internal and scoped to an organization, then you can only give access to other organization members or teams.

1. On GitHub, navigate to the main page of your organization.
2. Under your organization name, click the **Packages** tab.



3. Search for and then click the name of the package that you want to manage.
4. On your package's landing page, on the right-hand side, click **Package settings**.



5. Under "Manage access" or "Inherited access", click **Invite teams or people** and enter the name, username, or email of the person you want to give access. You can also enter a team name from the organization to give all team members access.
6. Next to the username or team name, use the **Role** drop-down menu to select a desired permission level.

The selected users or teams will automatically be given access and don't need to accept an invitation first.

Selecting whether a package inherits permissions from a repository

By default, if you publish a package that is linked to a repository, the package inherits the access permissions of the linked repository. We recommend you let packages inherit their permissions from a repository, because this simplifies the process of managing access to a package.

When a package inherits permissions from a repository, to grant or remove access to your package, you must configure the permissions of the linked repository.

If you want to configure a package's access settings on a granular level, separately from the linked repository, you must remove the inherited permissions from the package.

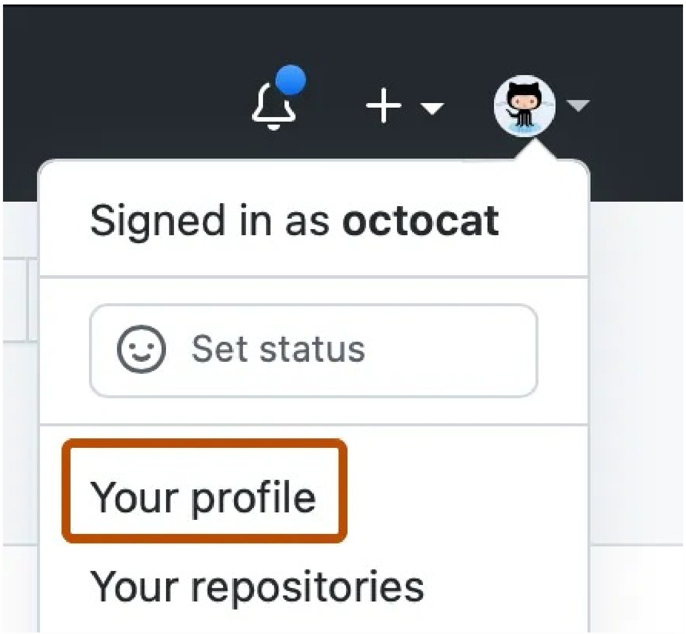
> **Note:** If you change how a package gets its access permissions, any existing permissions for the package are overwritten.

Selecting the inheritance setting for packages scoped to your

personal account

1. On GitHub, navigate to the main page of your personal account.

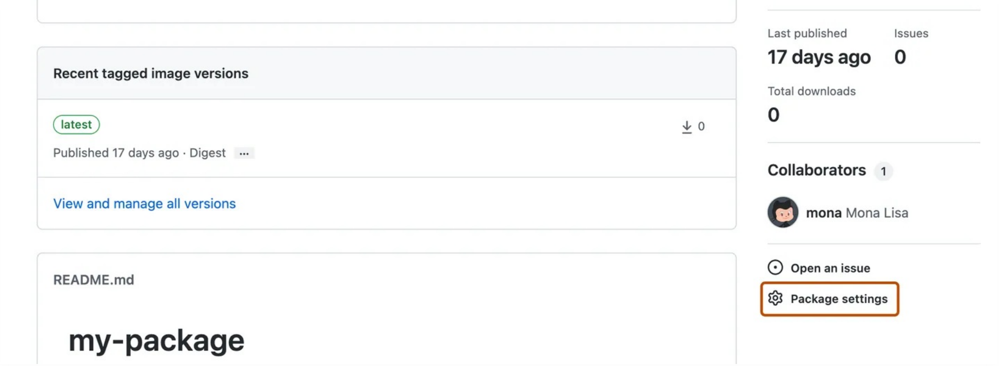
2. In the top right corner of GitHub.com, click your profile photo, then click **Your profile**.



3. On your profile page, in the header, click the **Packages** tab.

4. Search for and then click the name of the package that you want to manage.

5. On your package's landing page, on the right-hand side, click **Package settings**.



Recent tagged image versions

latest

Published 17 days ago · Digest ...

0

View and manage all versions

README.md

my-package

Last published 17 days ago

Issues 0

Total downloads 0

Collaborators 1

mona Mona Lisa

6. To choose whether a package inherits access permissions from the linked repository, under "Manage access" or "Inherited access", select or deselect **Inherit access from repository (recommended)**.

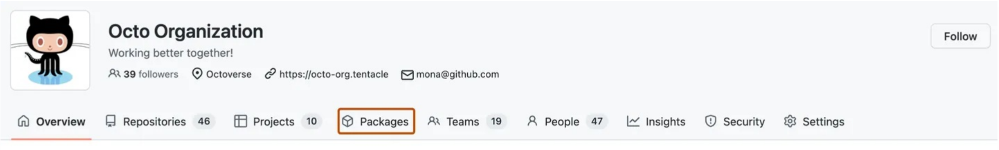
> **Note:** The name of this section changes depending on whether the package already inherits its permissions from a repository.

Selecting the inheritance setting for packages scoped to an organization

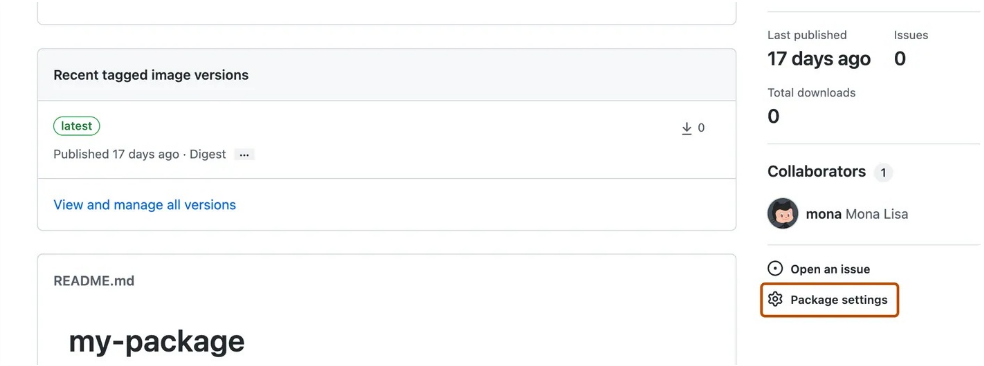
> **Tip:** If you're the owner of an organization, you can prevent all new packages scoped to your organization from automatically inheriting permissions from a linked repository. For more

information, see "Disabling automatic inheritance of access permissions in an organization" below.

1. On GitHub, navigate to the main page of your organization.
2. Under your organization name, click the **Packages** tab.



3. Search for and then click the name of the package that you want to manage.
4. On your package's landing page, on the right-hand side, click **Package settings**.



5. To choose whether a package inherits access permissions from the linked repository, under "Manage access" or "Inherited access", select or deselect **Inherit access from repository (recommended)**.

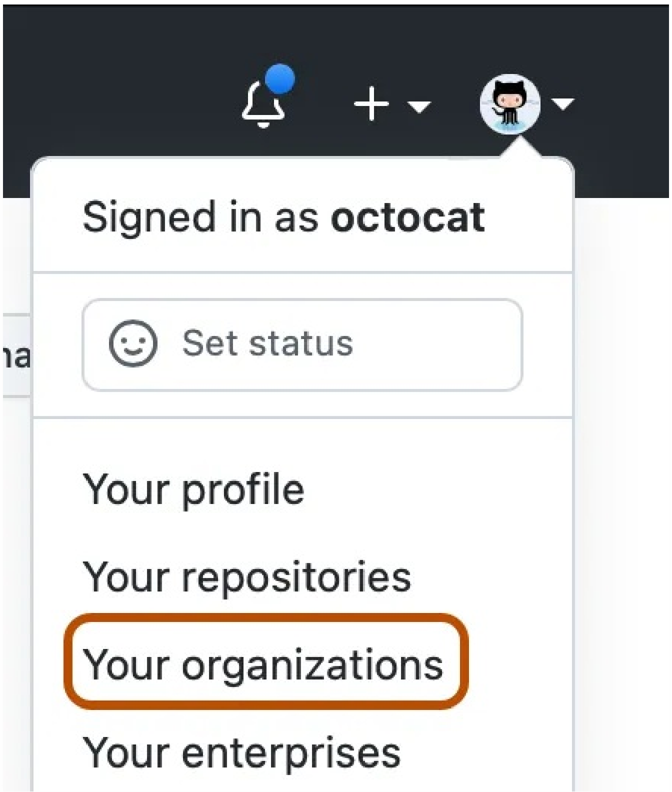
   > **Note:** The name of this section changes depending on whether the package already inherits its permissions from a repository.

Disabling automatic inheritance of access permissions in an organization

By default, if you publish a package that is linked to a repository, the package automatically inherits the access permissions of the linked repository. As an organization owner, you can disable automatic inheritance for all packages scoped to your organization.

If you disable automatic inheritance of access permissions, new packages scoped to your organization will not automatically inherit the permissions of a linked repository. However, anyone with admin permissions to a package in your organization will be able to enable or disable inheritance of permissions for that package.

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



Signed in as **octocat**

Set status

Your profile

Your repositories

Your organizations

Your enterprises

2. Next to the organization, click **Settings**.
3. In the sidebar, in the "Code, planning, and automation" section, click **Packages**.
4. Under "Default Package Settings", deselect **Inherit access from source repository**.
5. Click **Save**.

Ensuring workflow access to your package

For packages scoped to a personal account or an organization, to ensure that a GitHub Actions workflow has access to your package, you must give explicit access to the repository where the workflow is stored.

The specified repository does not need to be the repository where the source code for the package is kept. You can give multiple repositories workflow access to a package.

If you publish a package that is linked to a repository, GitHub Actions workflows in the linked repository automatically get access to the package, unless your organization has disabled the automatic inheritance of access permissions. For more information, see "About inheritance of access permissions and visibility" above.

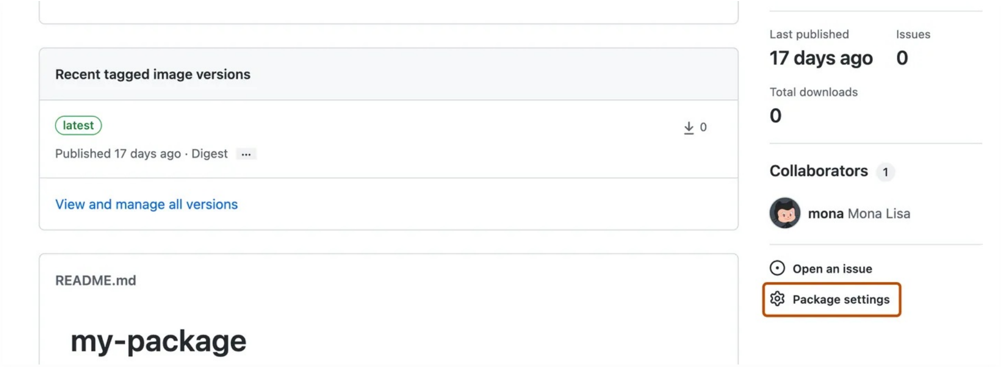
Notes:

- Syncing your package with a repository by using the **Add Repository** button under "Manage Actions access" in the package's settings is different than connecting your package to a repository. For more information about linking a repository to your package, see "Connecting a repository to a package."
- You can choose to limit permissions to workflow jobs usings the `permissions` key and `packages` scope. For more information, see "Assigning permissions to jobs."

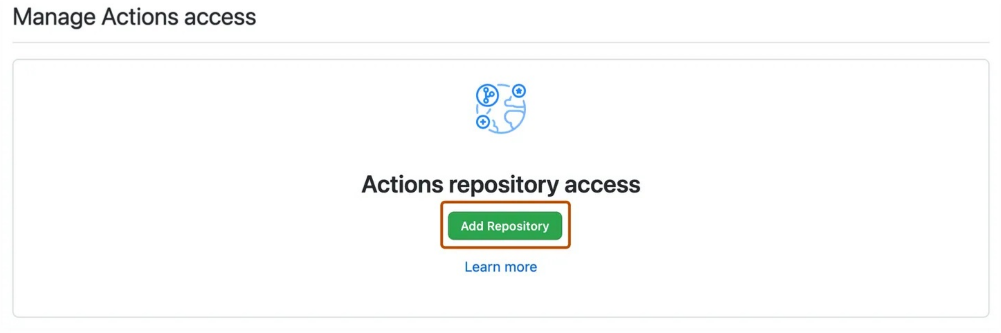
GitHub Actions access for packages scoped to personal accounts

1. Search for and then click the name of the package that you want to manage.

2. On your package's landing page, on the right-hand side, click ⚙ **Package settings**.



3. To ensure your workflow has access to your package, you must add the repository where the workflow is stored. Under "Manage Actions access", click **Add repository** and search for the repository you want to add.



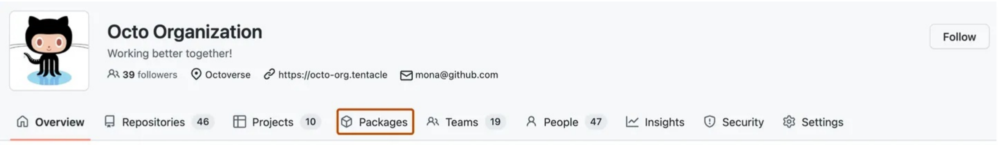
4. Use the **Role** drop-down menu to select the default access level that you'd like the repository to have to your package.

To further customize access to your package, see "[Configuring access to packages for your personal account](#)."

GitHub Actions access for packages scoped to organizations

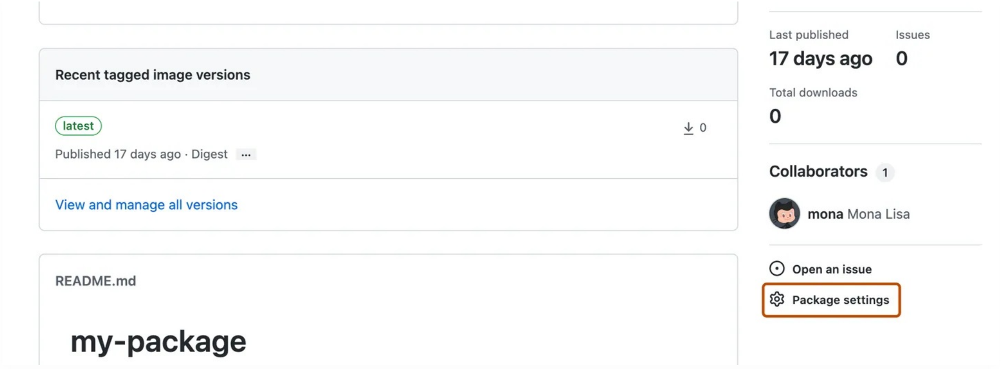
1. On GitHub, navigate to the main page of your organization.

2. Under your organization name, click the ⬡ **Packages** tab.



3. Search for and then click the name of the package that you want to manage.

4. On your package's landing page, on the right-hand side, click ⚙ **Package settings**.



Recent tagged image versions

latest

Published 17 days ago · Digest ...

0

View and manage all versions

README.md

my-package

Last published 17 days ago

Issues 0

Total downloads 0

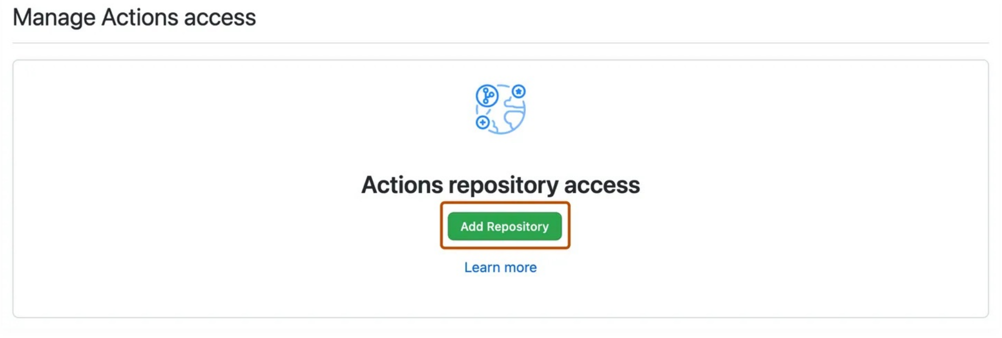
Collaborators 1

mona Mona Lisa

Open an issue

Package settings

5. Under "Manage Actions access", click **Add repository** and search for the repository you want to add.



Manage Actions access

Actions repository access

Add Repository

Learn more

6. Use the **Role** drop-down menu to select the default access level that you'd like the repository to have to your package.

To further customize access to your package, see "Configuring access to packages for an organization."

Ensuring GitHub Codespaces access to your package

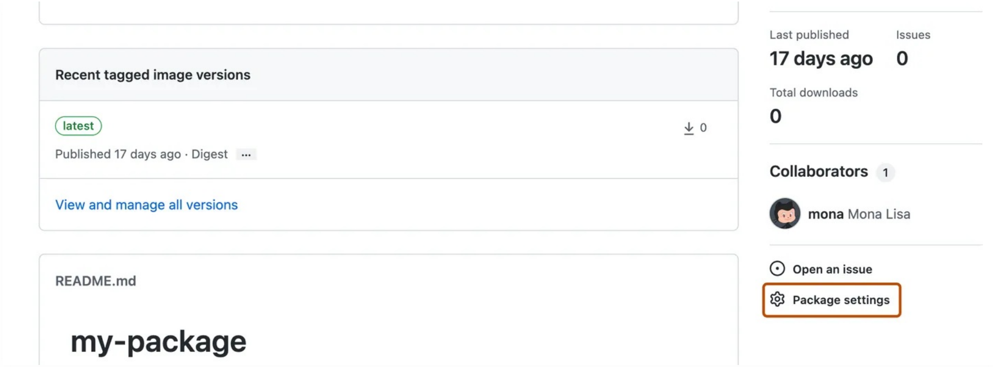
By default, a codespace can seamlessly access certain packages in registries that support granular permissions, such as packages published in the same repository with the **Inherit access** option selected. For the list of GitHub Packages registries that support granular permissions and seamless GitHub Codespaces access, see "About permissions for GitHub Packages."

Otherwise, to ensure that a codespace has access to your package, you must grant access to the repository where the codespace is being launched.

The specified repository does not need to be the repository where the source code for the package is kept. You can give codespaces in multiple repositories access to a package.

Once you've selected the package you're interested in sharing with codespaces in a repository, you can grant that repo access.

1. Search for and then click the name of the package that you want to manage.
2. On your package's landing page, on the right-hand side, click ⚙ **Package settings**.



Recent tagged image versions

latest

Published 17 days ago · Digest ...

↓ 0

View and manage all versions

README.md

my-package

Last published
17 days ago

Issues
0

Total downloads
0

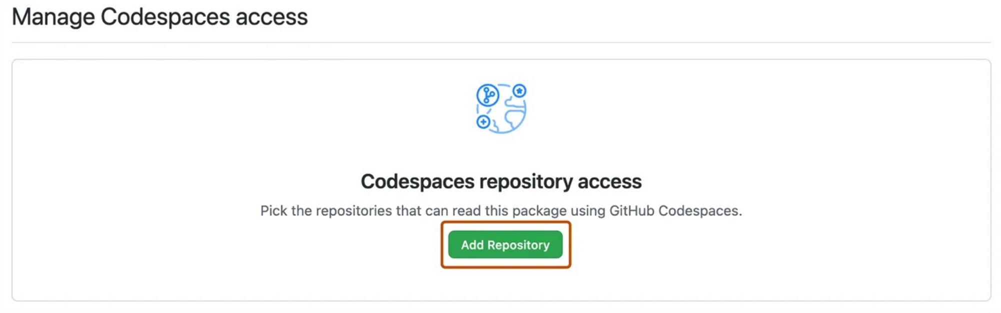
Collaborators 1

mona Mona Lisa

Open an issue

Package settings

3. Under "Manage Codespaces access", click **Add repository**.



Manage Codespaces access

Codespaces repository access

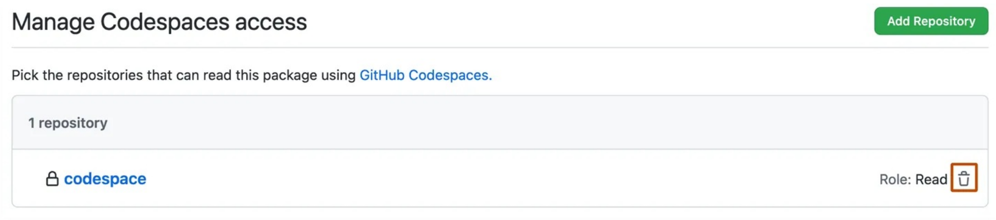
Pick the repositories that can read this package using GitHub Codespaces.

Add Repository

4. Search for the repository you want to add.

5. Repeat for any additional repositories you would like to allow access.

6. If the codespaces for a repository no longer need access to a package, you can remove access. Click 🗑.



Manage Codespaces access

Add Repository

Pick the repositories that can read this package using GitHub Codespaces.

1 repository

codespace

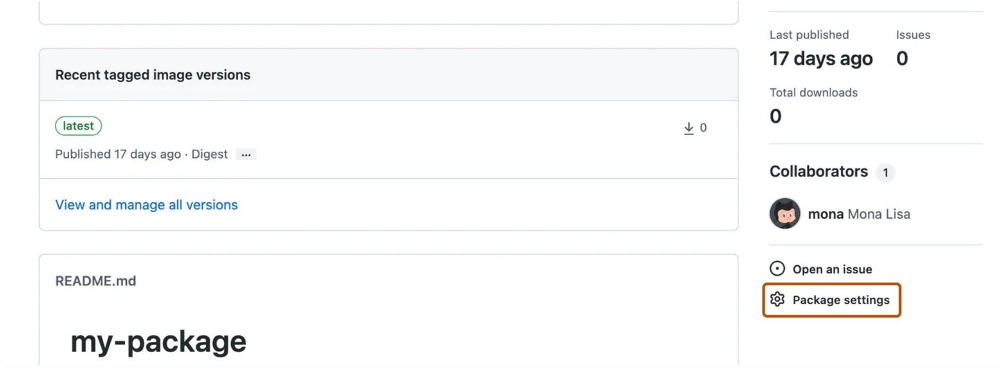
Role: Read

Configuring visibility of packages for your personal account

When you first publish a package that is scoped to your personal account, the default visibility is private and only you can see the package. You can modify a private or public package's access by changing the access settings. Once you make your package public, you cannot make your package private again.

1. Search for and then click the name of the package that you want to manage.

2. On your package's landing page, on the right-hand side, click ⚙ **Package settings**.



3. At the bottom of the page, under "Danger Zone", click **Change visibility**.

4. Select a visibility setting:

   - To make the package visible to anyone, select **Public**.

     > **Warning:** Once you make a package public, you cannot make it private again.

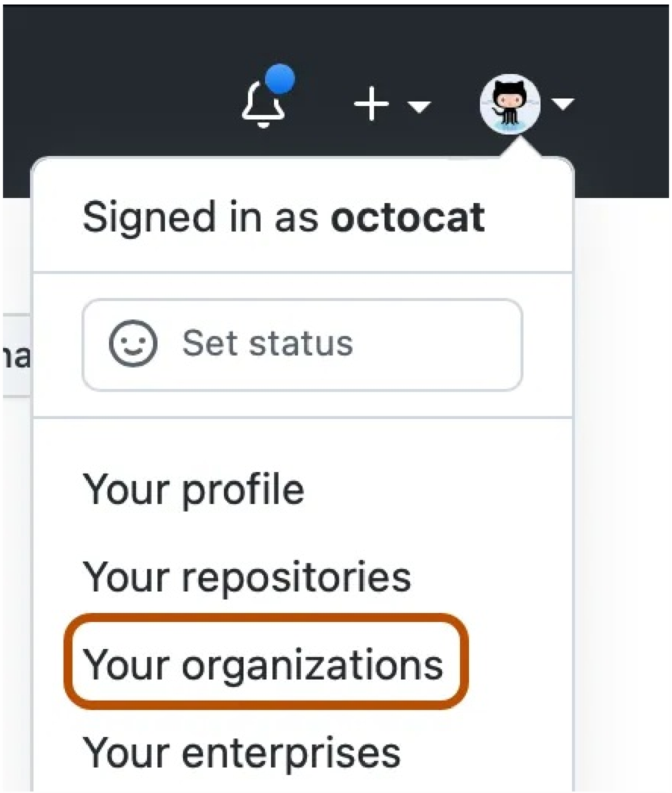
   - To make the package visible to a custom selection of people, select **Private**.

5. To confirm, type the name of the package, then click **I understand the consequences, change package visibility**.

Package creation visibility for organization members

For registries that support granular permissions, you can choose the visibility of packages that organization members can publish by default. For the list of these registries, see "About permissions for GitHub Packages."

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



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

3. On the left, click **Packages**.

4. Under "Package Creation", choose whether you want to enable the creation of public, private, or internal packages.

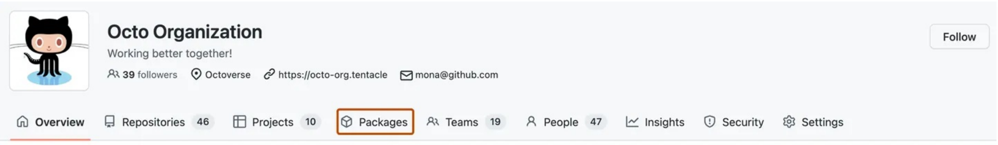
   - To enable organization members to create public packages, click **Public**.
   - To enable organization members to create private packages that are only visible to other organization members, click **Private**. You can further customize the visibility of private packages.
   - To enable organization members to create internal packages that are visible to all organization members, click **Internal**. If the organization belongs to an enterprise, the packages will be visible to all enterprise members.

Configuring visibility of packages for an organization

When you first publish a package, the default visibility is private and only you can see the package. You can grant users or teams different access roles for your package through the access settings. Once you make your package public, you cannot make your package private again.

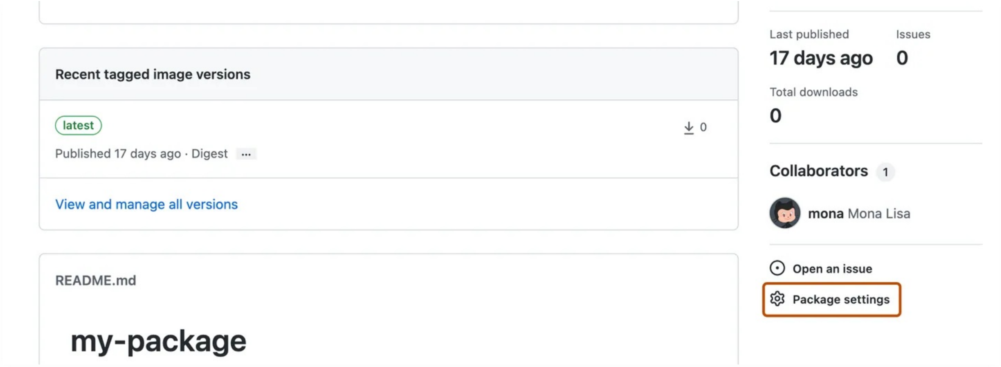
1. On GitHub, navigate to the main page of your organization.

2. Under your organization name, click the Packages tab.



3. Search for and then click the name of the package that you want to manage.

4. On your package's landing page, on the right-hand side, click ⚙ **Package settings**.



5. At the bottom of the page, under "Danger Zone", click **Change visibility** and choose a visibility setting:

   - To make the package visible to anyone, click **Public**.

     > **Warning:** Once you make a package public, you cannot make it private again.

   - To make the package visible to a custom selection of people in your organization, click **Private**.
   - To make the package visible to all organization members, click **Internal**. If the organization belongs to an enterprise, the packages will be visible to all enterprise members.