

# Managing secrets for your codespaces

## In this article

About secrets for GitHub Codespaces

Adding a secret

Editing a secret

Deleting a secret

Using secrets

Further reading

You can store sensitive information, like tokens, that you want to access in your codespaces via environment variables.

## About secrets for GitHub Codespaces [↗](#)

You can add secrets to your personal account that you want to use in your codespaces. For example, you may want to store and access the following sensitive information as secrets:

- Access tokens to cloud services
- Service principals
- Subscription identifiers
- Credentials for a private image registry (for more information, see "[Allowing your codespace to access a private registry](#)")

You can choose which repositories should have access to each secret. Then, you can use the secret in any codespace you create for a repository that has access to the secret. To share a secret with a codespace created from a template, you will need to publish the codespace to a repository on GitHub, then give that repository access to the secret.

Once you have created a secret, it will be available when you create a new codespace or restart the codespace. If you've created a secret on GitHub.com and you want to use it in a currently running codespace, stop the codespace and then restart it. For information about stopping the codespace, see "[Using the Visual Studio Code Command Palette in GitHub Codespaces](#)."

## Naming secrets [↗](#)

The following rules apply to secret names:

- Secret names can only contain alphanumeric characters ( `[a-z]` , `[A-Z]` , `[0-9]` ) or underscores ( `_` ). Spaces are not allowed.
- Secret names must not start with the `GITHUB_` prefix.
- Secret names must not start with a number.
- Secret names are not case-sensitive.
- Secret names must be unique at the level they are created at. For example, a secret created at the repository level must have a unique name in that repository.

If a secret with the same name exists at multiple levels, the secret at the lowest level

takes precedence. For example, if an organization-level secret has the same name as a repository-level secret, then the repository-level secret takes precedence.

## Limits for secrets [↗](#)

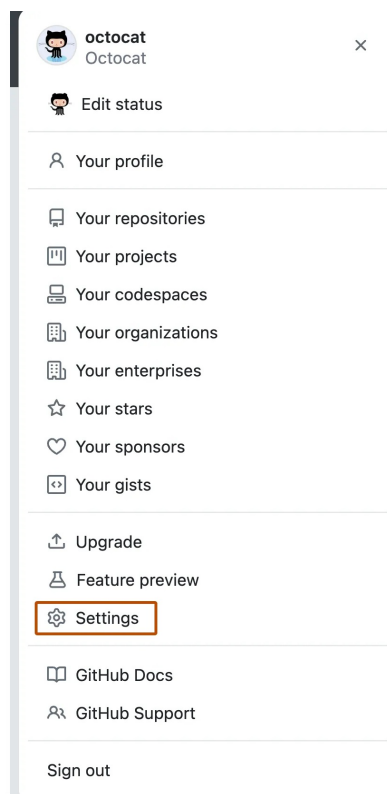
You can store up to 100 secrets for GitHub Codespaces.


Secrets are limited to 48 KB in size.

## Adding a secret [↗](#)

---

- 1 In the upper-right corner of any page, click your profile photo, then click **Settings**.



- 2 In the "Code, planning, and automation" section of the sidebar, click  **Codespaces**.
- 3 To the right of "Codespaces secrets", click **New secret**.
- 4 Under "Name," type a name for your secret.
- 5 Under "Value", type the value of your secret.
- 6 Select the "Repository access" drop-down menu, then click a repository you want to have access to the secret. Repeat for every repository you want to have access to the secret.

### Repository access

Select repositories ▼

Search for a repository

octocat/cat-repo

octo-org/octo-repo

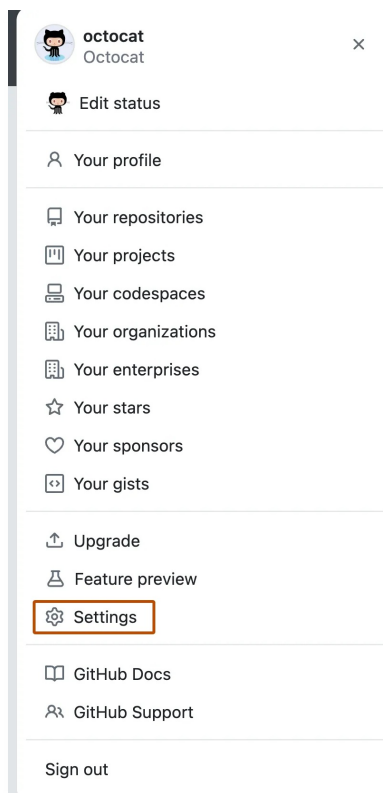
ive until at least 1 repository is selected.


- 7 Click **Add secret**.

## Editing a secret [↗](#)

You can update the value of an existing secret, and you can change which repositories can access a secret.

- 1 In the upper-right corner of any page, click your profile photo, then click **Settings**.



- 2 In the "Code, planning, and automation" section of the sidebar, click  **Codespaces**.
- 3 Under "Codespaces secrets," to the right of the secret you want to edit, click **Update**.
- 4 Under "Value," click the link "**enter a new value.**"

### GH\_TOKEN

#### Value

Secret values are encrypted and cannot be displayed, but you can [enter a new value.](#)

- 5 Under "Value", type the value of your secret.
- 6 Select the "Repository access" drop-down menu, then click a repository you want to have access to the secret. Repeat for every repository you want to have access to the secret.

#### Repository access

Select repositories ▼

Search for a repository

octocat/cat-repo

octo-org/octo-repo

ive until at least 1 repository is selected.

- 7 Optionally, to remove the secret's access to a repository, deselect the repository.

Select repositories ▼

Available to 2 repositories.

☒ octocat/cat-repo

☒ octo-org/octo-repo


Save changes

- 8 Click **Save changes**.

## Deleting a secret [↗](#)

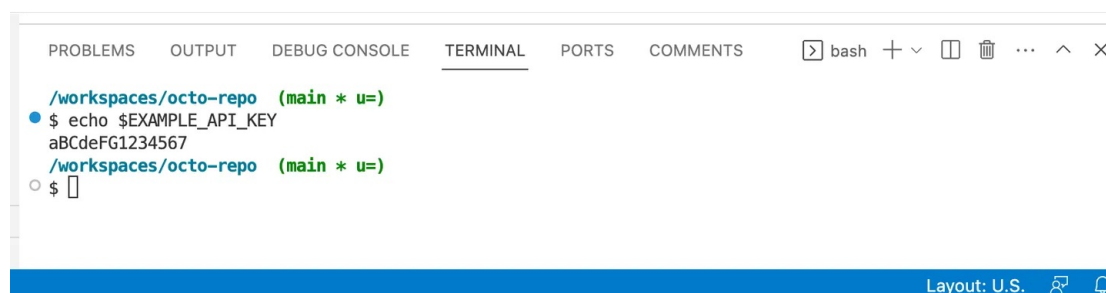
- 1 In the upper-right corner of any page, click your profile photo, then click **Settings**.



- 2 In the "Code, planning, and automation" section of the sidebar, click  **Codespaces**.
- 3 Under "Codespaces secrets," to the right of the secret you want to delete, click **Delete**.
- 4 Read the warning, then click **OK**.

## Using secrets

A secret is exported as an environment variable into the user's terminal session.



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS bash + - [ ] [ ] ... ^ X
/workspaces/octo-repo (main * u=)
$ echo $EXAMPLE_API_KEY
aBCdeFG1234567
/workspaces/octo-repo (main * u=)
$ 
```

You can use secrets in a codespace after the codespace is built and is running. For example, a secret can be used:

- When launching an application from the integrated terminal or ssh session.
- Within a dev container lifecycle script that is run after the codespace is running. For more information about dev container lifecycle scripts, see the documentation on the Development Containers website: [Specification](#).

Codespace secrets cannot be used:

- During codespace build time (that is, within a Dockerfile or custom entry point).
- Within a dev container feature. For more information, see the `features` property in the [dev containers specification](#) on the Development Containers website.

## Further reading

---

- "[Managing secrets for your repository and organization for GitHub Codespaces](#)"
- "[Creating a codespace for a repository](#)"

## Legal

© 2023 GitHub, Inc. [Terms](#) [Privacy](#) [Status](#) [Pricing](#) [Expert services](#) [Blog](#)