

Signing up for a new GitHub account

[About new accounts on GitHub.com](#)

You can create a personal account, which serves as your identity on GitHub.com, or an organization, which allows multiple personal accounts to collaborate across multiple projects. For more information about account types, see "[Types of GitHub accounts](#)."

When you create a personal account or organization, you must select a billing plan for the account. For more information, see "[GitHub's products](#)."

[Signing up for a new account](#)

1. If you want to create a new personal account, make sure you are currently signed out of GitHub.
2. Go to GitHub's [Pricing](#) page.
3. Read the information about the different products and subscriptions that GitHub offers, then click the upgrade button under the subscription you'd like to choose.
4. Follow the prompts to create your personal account or organization.

Verifying your email address

[About email verification](#)

You can verify your email address after signing up for a new account, or when you add a new email address. If an email address is undeliverable or bouncing, it will be unverified.

If you do not verify your email address, you will not be able to:

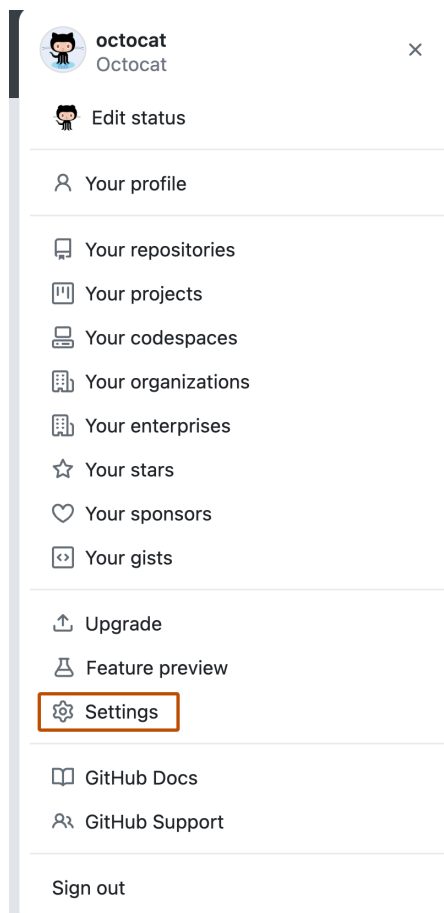
- Create or fork repositories
- Create issues or pull requests
- Comment on issues, pull requests, or commits
- Authorize OAuth app applications
- Generate personal access tokens
- Receive email notifications
- Star repositories
- Create or update project boards, including adding cards
- Create or update gists
- Create or use GitHub Actions
- Sponsor developers with GitHub Sponsors
- Accept organization invitations

Warnings:

- You cannot verify email addresses from disposable email address services (services that allow you to receive email at a temporary address that expires after a certain time). If you'd like to keep your email address private, you can use a GitHub-provided `noreply` email address. For more information, see "[Setting your commit email address](#)."
- If an organization you're a member of restricts email notifications to an approved email domain, you'll need to verify an email address in that domain to receive email notifications about activity in the organization. For more information, see "[Restricting email notifications for your organization](#)."

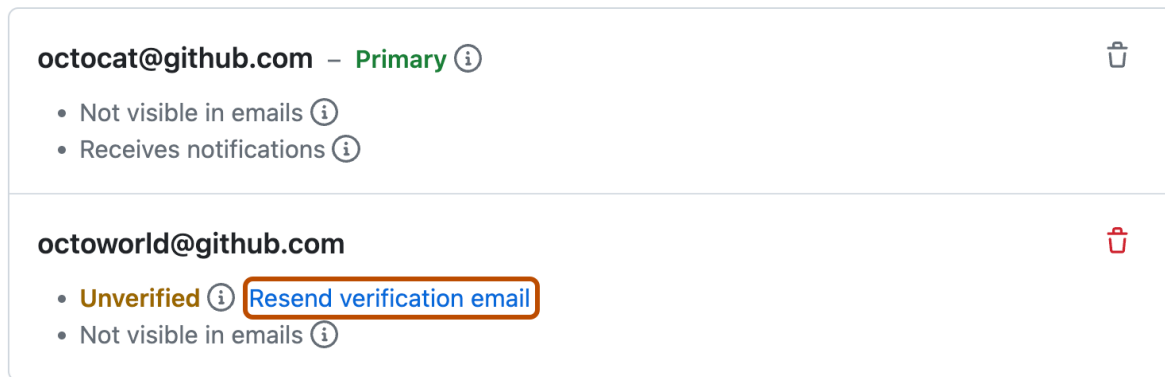
Verifying your email address

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



2. In the "Access" section of the sidebar, click **Emails**.
3. Under your email address, click **Resend verification email**.

Emails



4. GitHub will send you an email with a link in it. After you click that link, you'll be taken to your GitHub dashboard and see a confirmation banner.

Troubleshooting email verification

Unable to send verification email

You cannot verify email addresses from disposable email address services (services that allow you to receive email at a temporary address that expires after a certain time). If you'd like to keep your email address private, you can use a GitHub-provided `noreply` email address. For more information, see "[Setting your commit email address](#)."

Error page after clicking verification link

The verification link expires after 24 hours. If you don't verify your email within 24 hours, you can request another email verification link. For more information, see "[Verifying your email address](#)."

If you click on the link in the confirmation email within 24 hours and you are directed to an error page, you should ensure that you're signed into the correct account on GitHub.com.

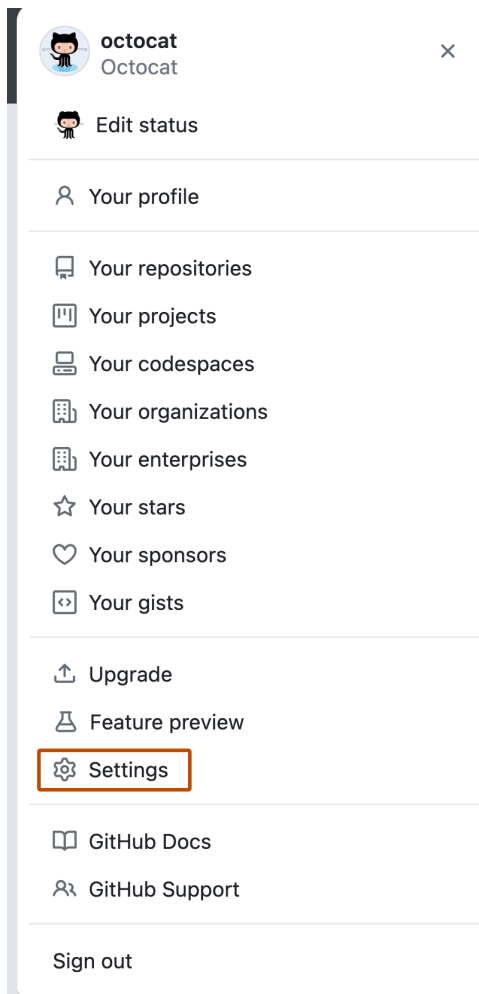
1. [Sign out](#) of your personal account on GitHub.com.
2. Quit and restart your browser.
3. [Sign in](#) to your personal account on GitHub.com.
4. Click on the verification link in the email we sent you.

Changing your primary email address

You can change the email address associated with your personal account at any time.

Note: You cannot change your primary email address to an email that is already set to be your backup email address.

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



2. In the "Access" section of the sidebar, click **Emails**.
3. If you'd like to add a new email address to set as your primary email address, under "Add email address", type a new email address and click **Add**.
4. Under "Primary email address", use the drop-down menu to click the email address you'd like to set as your primary email address, and click **Save**.
5. To remove the old email address from your account, next to the old email, click .
6. Verify your new primary email address. Without a verified email address, you won't be able to use all of GitHub's features. For more information, see "[Verifying your email address](#)."

Setting a backup email address

Use a backup email address as an additional destination for security-relevant account notifications and to securely reset your password if you can no longer access your primary email address.

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



octocat
Octocat



Edit status



Your profile



Your repositories



Your projects



Your codespaces



Your organizations



Your enterprises



Your stars



Your sponsors



Your gists



Upgrade



Feature preview



Settings



GitHub Docs



GitHub Support

Sign out

2. In the "Access" section of the sidebar, click **Emails**.
3. Under "Backup email address," use the dropdown menu to select the address you want to set as your backup email address.
4. Click **Save**.

Setting your commit email address

About commit email addresses

GitHub uses your commit email address to associate commits with your account on GitHub.com. You can choose the email address that will be associated with the commits you push from the command line as well as web-based Git operations you make.

For web-based Git operations, you can set your commit email address on GitHub.com. For commits you push from the command line, you can set your commit email address in Git.

Any commits you made prior to changing your commit email address are still associated with your previous email address.

Note: You cannot verify email addresses from disposable email address services (services that allow you to receive email at a temporary address that expires after a certain time). If you'd like to keep your email address private, you can use a GitHub-provided noreply email address. For more information, see "[Setting your commit email address](#)."

If you'd like to keep your personal email address private, you can use a noreply email address from GitHub as your commit email address. To use your noreply email address for commits you push from the command line, use that email address when you set your commit email address in Git. To use your noreply address for web-based Git operations, set your commit email address on GitHub and choose to **Keep my email address private**.

You can also choose to block commits you push from the command line that expose your personal email address. For more information, see "[Blocking command line pushes that expose your personal email address](#)."

To ensure that commits are attributed to you and appear in your contributions graph, use an email address that is connected to your account on GitHub.com, or the noreply email address provided to you in your email settings. For more information, see "[Adding an email address to your GitHub account](#)."

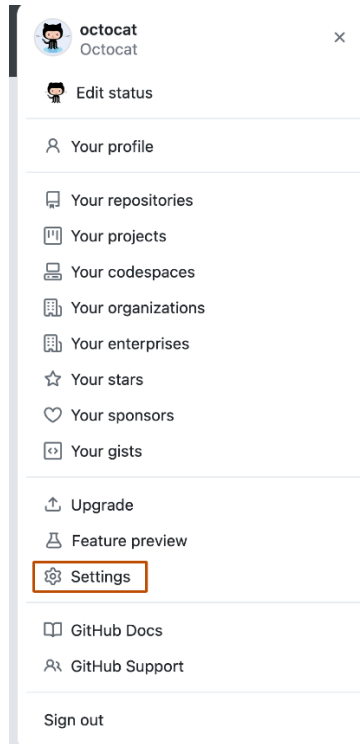
Note: If you created your account on GitHub.com *after* July 18, 2017, your noreply email address for GitHub is an ID number and your username in the form of [ID+USERNAME@users.noreply.github.com](#). If you created your account on GitHub.com *prior to* July 18, 2017, your noreply email address from GitHub is [USERNAME@users.noreply.github.com](#). You can get an ID-based noreply email address for GitHub by selecting (or deselecting and reselecting) **Keep my email address private** in your email settings.

If you use your noreply email address for GitHub to make commits and then [change your username](#), those commits will not be associated with your account on GitHub.com. This does not apply if you're using the ID-based noreply address from GitHub. For more information, see "[Changing your GitHub username](#)."

Setting your commit email address on GitHub

If you haven't enabled email address privacy, you can choose which verified email address to author changes with when you edit, delete, or create files or merge a pull request on GitHub. If you enabled email address privacy, then the commit author email address cannot be changed and is `<username>@users.noreply.github.com` by default.

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



2. In the "Access" section of the sidebar, click **Emails**.
3. In "Add email address", type your email address and click **Add**.
4. [Verify your email address](#).
5. In the "Primary email address" dropdown menu, select the email address you'd like to associate with your web-based Git operations.

Add email address

Primary email address

Because you have email privacy enabled, `octocat@github.com` will be used for account-related notifications as well as password resets. `101839405+octocat@users.noreply.github.com` will be used for web-based Git operations, e.g., edits and merges.

6. To keep your email address private when performing web-based Git operations, select **Keep my email addresses private**.

Setting your commit email address in Git

You can use the `git config` command to change the email address you associate with your Git commits. The new email address you set will be visible in any future commits you push to GitHub.com from the command line. Any commits you made prior to changing your commit email address are still associated with your previous email address.

Setting your email address for every repository on your computer

1. Open Git Bash.
2. Set an email address in Git. You can use your [GitHub-provided noreply email address](#) or any email address.

```
git config --global user.email "YOUR_EMAIL"
```
3. Confirm that you have set the email address correctly in Git:
4.

```
$ git config --global user.email  
email@example.com
```
5. Add the email address to your account on GitHub, so that your commits are attributed to you and appear in your contributions graph. For more information, see "[Adding an email address to your GitHub account](#)."

Setting your email address for a single repository

GitHub uses the email address set in your local Git configuration to associate commits pushed from the command line with your account on GitHub.com.

You can change the email address associated with commits you make in a single repository. This will override your global Git configuration settings in this one repository, but will not affect any other repositories.

1. Open Git Bash.
2. Change the current working directory to the local repository where you want to configure the email address that you associate with your Git commits.
3. Set an email address in Git. You can use your [GitHub-provided noreply email address](#) or any email address.

```
git config user.email "YOUR_EMAIL"
```
4. Confirm that you have set the email address correctly in Git:
5.

```
$ git config user.email  
email@example.com
```
6. Add the email address to your account on GitHub, so that your commits are attributed to you and appear in your contributions graph. For more information, see "[Adding an email address to your GitHub account](#)."

Remembering your GitHub username or email

GitHub Desktop users

1. In the **File** menu, click **Options**.
2. In the Options window, verify the following:
 - To view your GitHub username, click **Accounts**.
 - To view your Git email, click **Git**. Note that this email is not guaranteed to be [your primary GitHub email](#).

Finding your username in your `user.name` configuration

During set up, you may have [set your username in Git](#). If so, you can review the value of this configuration setting:

```
$ git config user.name
# View the setting
YOUR_USERNAME
```

Finding your username in the URL of remote repositories

If you have any local copies of personal repositories you have created or forked, you can check the URL of the remote repository.

Tip: This method only works if you have an original repository or your own fork of someone else's repository. If you clone someone else's repository, their username will show instead of yours. Similarly, organization repositories will show the name of the organization instead of a particular user in the remote URL.

```
$ cd YOUR_REPOSITORY
# Change directories to the initialized Git repository
$ git remote -v
origin  https://github.com/YOUR_USERNAME/YOUR_REPOSITORY.git (fetch)
origin  https://github.com/YOUR_USERNAME/YOUR_REPOSITORY.git (push)
Your user name is what immediately follows the https://github.com/.
```

Inviting collaborators to a personal repository

You can invite users to become collaborators to your personal repository.

If you're using GitHub Free, you can add unlimited collaborators on public and private repositories.

[About collaboration in a personal repository](#)

To collaborate with users in a repository that belongs to your personal account on GitHub.com, you can invite the users as collaborators.

If you want to grant more granular access to the repository, you can create a repository within an organization. For more information, see "[Access permissions on GitHub.](#)"

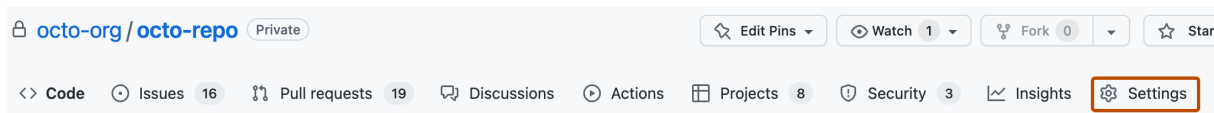
Private forks inherit the permissions structure of the upstream repository. This helps owners of private repositories maintain control over their code. For example, if the upstream repository is private and gives read/write access to a team, then the same team will have read/write access to any forks of the private upstream repository. Only team permissions (not individual permissions) are inherited by private forks.

[Inviting a collaborator to a personal repository](#)

You can send an invitation to collaborate in your repository directly to someone on GitHub.com, or to the person's email address

GitHub limits the number of people who can be invited to a repository within a 24-hour period. If you exceed this limit, either wait 24 hours or create an organization to collaborate with more people. For more information, see "[Creating a new organization from scratch.](#)"

1. Ask for the username of the person you're inviting as a collaborator. If they don't have a username yet, they can sign up for GitHub. For more information, see "[Signing up for a new GitHub account.](#)"
2. On GitHub.com, navigate to the main page of the repository.
3. Under your repository name, click **Settings**. If you cannot see the "Settings" tab, select the dropdown menu, then click **Settings**.



4. In the "Access" section of the sidebar, click **Collaborators**.
5. Click **Add people**.
6. In the search field, start typing the name of person you want to invite, then click a name in the list of matches.
7. Click **Add NAME to REPOSITORY**.

8. The user will receive an email inviting them to the repository. Once they accept your invitation, they will have collaborator access to your repository.

Removing a collaborator from a personal repository

When you remove a collaborator from your project, they lose read/write access to your repository. If the repository is private and the person has created a fork, then that fork is also deleted.

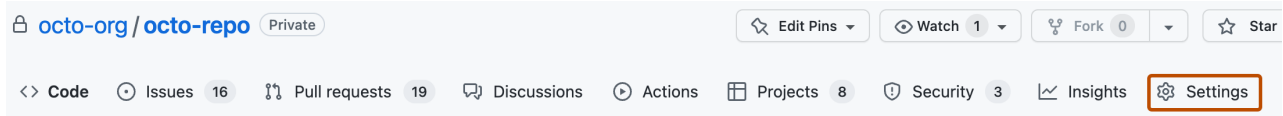
If you're using GitHub Free, you can add unlimited collaborators on public and private repositories.

[Deleting forks of private repositories](#)

While forks of private repositories are deleted when a collaborator is removed, the person will still retain any local clones of your repository.

[Removing collaborator permissions from a person contributing to a repository](#)

1. On GitHub.com, navigate to the main page of the repository.
2. Under your repository name, click **Settings**. If you cannot see the "Settings" tab, select the dropdown menu, then click **Settings**.

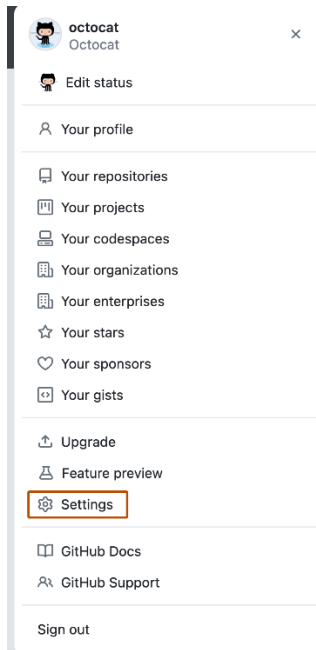


3. In the "Access" section of the sidebar, click **Collaborators & teams**.
4. To the right of the collaborator you want to remove, click **Remove**.

Removing yourself from a collaborator's repository

If you no longer want to be a collaborator on someone else's repository, you can remove yourself.

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 **Repositories**.
3. Next to the repository you want to leave, click **Leave**.
4. Read the warning carefully, then click **I understand, leave this repository**.

Maintaining ownership continuity of your personal account's repositories

You can invite someone to manage your user owned repositories if you are not able to.

[About successors](#)

We recommend inviting another GitHub user to be your successor, to manage your user owned repositories if you cannot. As a successor, they will have permission to:

- Archive your public repositories.
- Transfer your public repositories to their own user owned account.
- Transfer your public repositories to an organization where they can create repositories.

Successors cannot log into your account.

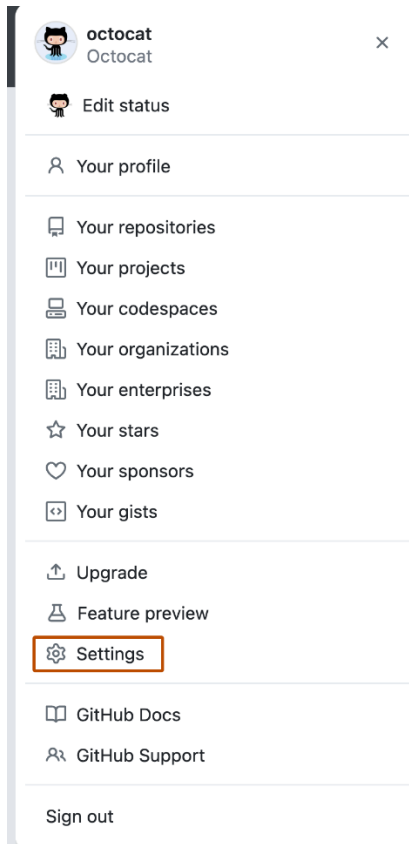
An appointed successor can manage your public repositories after presenting a death certificate then waiting for 7 days or presenting an obituary then waiting for 21 days. For more information, see "[GitHub Deceased User Policy](#)."

To request access to manage repositories as a successor, contact [GitHub Support](#).

[Inviting a successor](#)

The person you invite to be your successor must have a GitHub account.

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



2. In the left sidebar, click **Account**.
3. Under "Successor settings", to invite a successor, begin typing a username, full name, or email address, then click their name when it appears.

Successor settings

By clicking "Add Successor" below, I acknowledge that I am the owner of the @isaacmbrown account, and am authorizing GitHub to transfer content within that account to my GitHub Successor, designated below, in the event of my death. I understand that this appointment of a successor does not override legally binding next-of-kin rules or estate laws of any relevant jurisdiction, and does not create a binding will. [Learn more about account successors.](#)

Search by username, full name, or email address

A search result for the username 'octocat'. It shows a small profile picture of the user 'octocat' followed by the text 'octocat The Octocat'.

4. Click **Add successor**.
5. If prompted, confirm access to your account on GitHub. For more information, see "[Sudo mode](#)."

The user you've invited will be listed as "Pending" until they agree to become your successor.