

GitHub glossary

In this article

[@mention](#)

[access token](#)

[API preview](#)

[appliance](#)

[assignee](#)

[authentication code](#)

[base branch](#)

[basic authentication](#)

[billing cycle](#)

[billing email](#)

[billing manager](#)

[billing plan](#)

[bio](#)

[blame](#)

[block](#)

[branch](#)

[branch restriction](#)

[Business plan](#)

[CA certificate](#)

[card](#)

[check](#)

[checkout](#)

[cherry-picking](#)

[child team](#)

[clean](#)

[clone](#)

[clustering](#)

[code frequency graph](#)

[code of conduct](#)

[code owner](#)

[collaborator](#)

[commit](#)

[commit author](#)

[Commit graph](#)

[commit ID](#)

[commit message](#)

[compare branch](#)

[continuous integration](#)

[contribution graph](#)

[contribution guidelines](#)

[contributions](#)

[contributor](#)

[contributors graph](#)

coupon
cron
cURL
dashboard
default branch
dependency graph
dependents graph
deploy key
detached HEAD
diagnostics
diff
directory
dirty
email notifications
enterprise account
Explorer
fast-forward
feature branch
fenced code block
fetch
following (users)
force push
fork
Free plan
gist
Git
gitfile
GitHub App
GitHub Flavored Markdown
GitHub Importer
GitHub Jobs
GitHub Marketplace
GitHub Pages
GitHub Wiki
GraphQL
HEAD
head branch
Hello, World
high-availability
hook
hostname
identicon
identity provider
instance
integration
issue
Jekyll
Jekyll Theme Chooser
key fingerprint
keychain
keyword

label

LFS

license

line comment

line ending

Linguist

locked personal account

main

management console

Markdown

markup

master

members graph

mention

merge

merge conflict

milestone

mirror

nested team

network graph

news feed

non-fast-forward

notification

OAuth app

OAuth token

open source

organization

organization owner

origin

outside collaborator

owner

parent team

participating notifications

permalink

personal account

pinned repository

pre-receive hooks

primary email address

private contributions

private repository

production branch

profile

profile photo

project board

protected branch

public contributions

public repository

pull

pull access

pull request

pull request review

pulse graph
punch graph
push
push a branch
push access
read access
README
rebase
recovery code
release
remote
remote repository
remote URL
replica
repository
repository cache
repository graph
repository maintainer
required pull request review
required status check
resolve
revert
review
root directory
root filesystem
saved reply
scope
seat
secret team
security log
server-to-server request
service hook
single sign-on
snapshot
squash
SSH key
staging instance
star
status
status checks
subscription
team
team maintainer
Team plan
timeline
topic branch
topics
traffic graph
transfer
upstream
upstream branch

- user
- user-to-server request
- username
- visible team
- watch
- watching notifications
- web notifications
- webhooks
- write access
- Further reading

This glossary introduces common Git and GitHub terminology.

@mention [↗](#)

To notify a person on GitHub by using `@` before their username. Users in an organization on GitHub can also be a part of a team that can be mentioned.

access token [↗](#)

A token that is used in place of a password when performing Git operations over HTTPS with Git on the command line or the API. Also called a personal access token.

API preview [↗](#)

A way to try out new APIs and changes to existing API methods before they become part of the official GitHub API.

appliance [↗](#)

A software application combined with just enough operating system (JeOS) to run optimally on industry-standard hardware (typically a server) or in a virtual machine.

assignee [↗](#)

The user that is assigned to an issue.

authentication code [↗](#)

A code you'll supply, in addition to your GitHub password, when signing in with 2FA via the browser. This code is either generated by an application or delivered to your phone via text message. Also called a "2FA authentication code."

base branch [↗](#)

The branch into which changes are combined when you merge a pull request. When you create a pull request, you can change the base branch from the repository's default branch to another branch if required.

basic authentication

A method of authentication where the credentials are sent as unencrypted text.

billing cycle

The interval of time for your specific billing plan.

billing email

The organization email address where GitHub sends receipts, credit card or PayPal charges, and other billing-related communication.

billing manager

The organization member that manages billing settings for an organization.

billing plan

Payment plans for users and organizations that include set features for each type of plan.

bio

The user-generated description found on a profile: [Adding a bio to your profile](#)

blame

The "blame" feature in Git describes the last modification to each line of a file, which generally displays the revision, author and time. This is helpful, for example, in tracking down when a feature was added, or which commit led to a particular bug.

block

To remove a user's ability to collaborate on an organization's repositories.

branch

A branch is a parallel version of a repository. It is contained within the repository, but does not affect the primary or main branch allowing you to work freely without disrupting the "live" version. When you've made the changes you want to make, you can merge your branch back into the main branch to publish your changes.

branch restriction

A restriction that repository admins can enable so that only certain users or teams can push or make certain changes to the branch.

Business plan

An organization billing plan where you can collaborate on unlimited public and private repositories, allow or require organization members to authenticate to GitHub using SAML SSO, and provision and deprovision access with SAML or SCIM.

CA certificate

A digital certificate issued by Certificate Authority (CA) that ensures there are valid connections between two machines, such as a user's computer and GitHub.com and verifies the ownership of a site.

card

A movable square within a project board associated with an issue or pull request.

check

A check is a type of status check on GitHub Enterprise Server. See "[Status checks](#)."

checkout

You can use `git checkout` on the command line to create a new branch, change your current working branch to a different branch, or even to switch to a different version of a file from a different branch with `git checkout [branchname] [path to file]`. The "checkout" action updates all or part of the working tree with a tree object or blob from the object database, and updates the index and HEAD if the whole working tree is pointing to a new branch.

cherry-picking

To choose a subset of changes from a series of changes (typically commits) and record them as a new series of changes on top of a different codebase. In Git, this is performed by the `git cherry-pick` command to extract the change introduced by an existing commit on another branch and to record it based on the tip of the current branch as a new commit. For more information, see [git-cherry-pick](#) in the Git documentation.

child team

Within nested teams, the subteam that inherits the parent team's access permissions and @mentions.

clean

A working tree is clean if it corresponds to the revision referenced by the current HEAD. Also see "[dirty](#)".

clone

A clone is a copy of a repository that lives on your computer instead of on a website's server somewhere, or the act of making that copy. When you make a clone, you can edit the files in your preferred editor and use Git to keep track of your changes without having to be online. The repository you cloned is still connected to the remote version so that you can push your local changes to the remote to keep them synced when you're

online.

clustering [↗](#)

The ability to run GitHub Enterprise services across multiple nodes and load balance requests between them.

code frequency graph [↗](#)

A repository graph that shows the content additions and deletions for each week in a repository's history.

code of conduct [↗](#)

A document that defines standards for how to engage in a community.

code owner [↗](#)

A person who is designated as an owner of a portion of a repository's code. The code owner is automatically requested for review when someone opens a pull request (not in draft mode) that makes changes to code the code owner owns.

collaborator [↗](#)

A collaborator is a person with read and write access to a repository who has been invited to contribute by the repository owner.

commit [↗](#)

A commit, or "revision", is an individual change to a file (or set of files). When you make a commit to save your work, Git creates a unique ID (a.k.a. the "SHA" or "hash") that allows you to keep record of the specific changes committed along with who made them and when. Commits usually contain a commit message which is a brief description of what changes were made.

commit author [↗](#)

The user who makes the commit.

Commit graph [↗](#)

A repository graph that shows all the commits made to a repository in the past year.

commit ID [↗](#)

Also known as SHA. A 40-character checksum hash that identifies the commit.

commit message [↗](#)

Short, descriptive text that accompanies a commit and communicates the change the

commit is introducing.

compare branch

The branch you use to create a pull request. This branch is compared to the base branch you choose for the pull request, and the changes are identified. When the pull request is merged, the base branch is updated with the changes from the compare branch. Also known as the "head branch" of the pull request.

continuous integration

Also known as CI. A process that runs automated builds and tests once a person commits a change to a configured repository on GitHub. CI is a common best practice in software development that helps detect errors.

contribution graph

The part of a user's profile that shows their contributions over a period of up to one year, day by day.

contribution guidelines

A document explaining how people should contribute to your project.

contributions

Specific activities on GitHub that will:

- Add a square to a user's contribution graph: "[Viewing contributions on your profile](#)"
- Add activities to a user's timeline on their profile: "[Viewing contributions on your profile](#)"

contributor

A contributor is someone who does not have collaborator access to a repository but has contributed to a project and had a pull request they opened merged into the repository.

contributors graph

A repository graph that displays the top 100 contributors to a repository.

coupon

A GitHub-provided code that users or organizations can use to pay for all or part of their subscription.

cron

A time-based job scheduler in Unix-like computer operating systems.

cURL

Used in command lines or scripts to transfer data.

dashboard

Your personal dashboard is the main hub of your activity on GitHub. From your personal dashboard, you can keep track of issues and pull requests you're following or working on, navigate to your top repositories and team pages, and learn about recent activity in repositories you're watching or participating in. You can also discover new repositories, which are recommended based on users you're following and repositories you have starred. To only view activity for a specific organization, visit your organization's dashboard. For more information, see "[About your personal dashboard](#)" or "[About your organization dashboard](#)."

default branch

The base branch for new pull requests and code commits in a repository. Each repository has at least one branch, which Git creates when you initialize the repository. The first branch is usually called `main`, and is often the default branch.

dependency graph

A repository graph that shows the packages and projects that the repository depends on.

dependents graph

A repository graph that shows the packages, projects, and repositories that depend on a public repository.

deploy key

A deploy key is an SSH key that is stored on your server and grants access to a single GitHub repository. This key is attached directly to the repository instead of to a personal user account.

detached HEAD

Git will warn you if you're working on a detached HEAD, which means that Git is not pointing to a branch and that any commits you make will not appear in commit history. For example, when you check out an arbitrary commit that is not the latest commit of any particular branch, you're working on a "detached HEAD."

diagnostics

An overview of a GitHub Enterprise instance's settings and environment.

diff

A diff is the difference in changes between two commits, or saved changes. The diff will visually describe what was added or removed from a file since its last commit.

directory

A folder containing one or more files or folders. You can create directories to organize the contents of a repository.

dirty

A working tree is considered "dirty" if it contains modifications that have not been committed to the current branch.

email notifications

Notifications sent to a user's email address.

enterprise account

Enterprise accounts allow you to centrally manage policy and billing for multiple organizations. Enterprise accounts are available with GitHub Enterprise Cloud and GitHub Enterprise Server. For more information, see "[About enterprise accounts](#)."

Explorer

An instance of GraphQL, which is a "graphical interactive in-browser GraphQL IDE."

fast-forward

A fast-forward is a special type of merge where you have a revision and you are "merging" another branch's changes that happen to be a descendant of what you have. In such a case, you do not make a new merge commit but instead just update to this revision. This will happen frequently on a remote-tracking branch of a remote repository.

feature branch

A branch used to experiment with a new feature or fix an issue that is not in production. Also called a topic branch.

fenced code block

An indented block of code you can create with GitHub Flavored Markdown using triple backticks ````` before and after the code block. See this [example](#).

fetch

When you use `git fetch`, you're adding changes from the remote repository to your local working branch without committing them. Unlike `git pull`, fetching allows you to review changes before committing them to your local branch.

following (users)

To get notifications about another user's contributions and activity.

force push

A Git push that overwrites the remote repository with local changes without regard for conflicts.

fork

A fork is a personal copy of another user's repository that lives on your account. Forks allow you to freely make changes to a project without affecting the original upstream repository. You can also open a pull request in the upstream repository and keep your fork synced with the latest changes since both repositories are still connected.

Free plan

A user account billing plan that is free. Users can collaborate on unlimited public repositories with unlimited collaborators.

gist

A gist is a shareable file that you can edit, clone, and fork on GitHub. You can make a gist public or secret, although secret gists will be available to anyone with the URL.

Git

Git is an open source program for tracking changes in text files. It was written by the author of the Linux operating system, and is the core technology that GitHub, the social and user interface, is built on top of.

gitfile

A plain `.git` file, which is always at the root of a working tree and points to the Git directory, which has the entire Git repository and its meta data. You can view this file for your repository on the command line with `git rev-parse --git-dir`. that is the real repository.

GitHub App

GitHub Apps provide a service to an entire organization and use their own identity when performing their function. They can be installed directly on organizations and user accounts and granted access to specific repositories. They come with granular permissions and built-in webhooks.

GitHub Flavored Markdown

GitHub-specific Markdown used to format prose and code across GitHub. See [GitHub Flavored Markdown Spec](#) or [Getting started with writing and formatting on GitHub](#).

GitHub Importer

A tool that quickly imports source code repositories, including commits and revision history, to GitHub for users.

GitHub Jobs [↗](#)

A GitHub site where employers can post jobs that GitHub users may be interested in.

GitHub Marketplace [↗](#)

A subsite for GitHub users and organizations to purchase and install applications that extend and complement their workflow.

GitHub Pages [↗](#)

Also referred to as Pages. A static site hosting service designed to host your personal, organization, or project pages directly from a GitHub repository.

GitHub Wiki [↗](#)

A section for hosting wiki style documentation on a GitHub repository.

GraphQL [↗](#)

A query language for APIs and a runtime for fulfilling those queries with your existing data.

HEAD [↗](#)

A defined commit of a branch, usually the most recent commit at the tip of the branch.

head branch [↗](#)

The branch whose changes are combined into the base branch when you merge a pull request. Also known as the "compare branch."

Hello, World [↗](#)

A "Hello, World!" program is a computer program that outputs or displays "Hello, World!" to a user. Since this program is usually very simple, it is often used as an example of a programming language's basic syntax and serves as a common first exercise for learning a new programming language.

high-availability [↗](#)

A system or component that is continuously operational for a desirably long length of time.

hook [↗](#)

During the normal execution of several Git commands, call-outs are made to optional scripts that allow a developer to add functionality or checking. Typically, the hooks allow for a command to be pre-verified and potentially aborted, and allow for a post-notification after the operation is done.

hostname

Human-readable nicknames that correspond to the address of a device connected to a network.

identicon

An auto-generated image used as a default profile photo when users sign up for GitHub. Users can replace their identicon with their own profile photo.

identity provider

Also known as an IdP. A trusted provider that lets you use SAML single sign-on (SSO) to access other websites.

instance

An organization's private copy of GitHub contained within a virtual machine that they configure and control.

integration

A third-party application that integrates with GitHub. These are often GitHub Apps, GitHub Actions, or custom actions. For more information, see "[About building integrations](#)."

issue

Issues are suggested improvements, tasks or questions related to the repository. Issues can be created by anyone (for public repositories), and are moderated by repository collaborators. Each issue contains its own discussion thread. You can also categorize an issue with labels and assign it to someone.

Jekyll

A static site generator for personal, project, or organization sites.

Jekyll Theme Chooser

An automated way to select a visual theme for your Jekyll site without editing or copying CSS files.

key fingerprint

A short sequence of bytes used to identify a longer public key.

keychain

A password management system in macOS.

keyword [↗](#)

A specific word that closes an issue when used within a pull request.

label [↗](#)

A tag on an issue or pull request. Repositories come with a handful of default labels, but users can create custom labels.

LFS [↗](#)

Git Large File Storage. An open source Git extension for versioning large files.

license [↗](#)

A document that you can include with your project to let people know what they can and can't do with your source code.

line comment [↗](#)

A comment within a pull request on a specific line of code.

line ending [↗](#)

An invisible character or characters that symbolize the end of a line in a text file.

Linguist [↗](#)

A library used on GitHub to detect blob languages, ignore binary or vendored files, suppress generated files in diffs, and generate language breakdown graphs.

locked personal account [↗](#)

A personal account that cannot be accessed by the user. Accounts are locked when users downgrade their paid account to a free one, or if their paid plan is past due.

main [↗](#)

The default development branch. Whenever you create a Git repository, a branch named `main` is created, and becomes the active branch. In most cases, this contains the local development, though that is purely by convention and is not required.

management console [↗](#)

A section within the GitHub Enterprise interface that contains administrative features.

Markdown [↗](#)

Markdown is an incredibly simple semantic file format, not too dissimilar from .doc, .rtf

and .txt. Markdown makes it easy for even those without a web-publishing background to write prose (including with links, lists, bullets, etc.) and have it displayed like a website. GitHub supports Markdown and uses a particular form of Markdown called GitHub Flavored Markdown. See [GitHub Flavored Markdown Spec](#) or [Getting started with writing and formatting on GitHub](#).

markup

A system for annotating and formatting a document.

master

The default branch in many Git repositories. By default, when you create a new Git repository on the command line, a branch called `master` is created. Many tools now use an alternative name for the default branch. For example, when you create a new repository on GitHub, the default branch is called `main`.

members graph

A repository graph that shows all the forks of a repository.

mention

A notification sent to a user by prefacing their username with the @ symbol. Users in an organization on GitHub can also be a part of a team that can be mentioned.

merge

Merging takes the changes from one branch (in the same repository or from a fork), and applies them into another. This often happens as a "pull request" (which can be thought of as a request to merge), or via the command line. A merge can be done through a pull request via the GitHub.com web interface if there are no conflicting changes, or can always be done via the command line.

merge conflict

A difference that occurs between merged branches. Merge conflicts happen when people make different changes to the same line of the same file, or when one person edits a file and another person deletes the same file. The merge conflict must be resolved before you can merge the branches.

milestone

A way to track the progress on groups of issues or pull requests in a repository.

mirror

A new copy of a repository.

nested team

A child team of a parent team. You can have multiple children (or nested) teams.

network graph

A repository graph that shows the branch history of the entire repository network, including branches of the root repository and branches of forks that contain commits unique to the network.

news feed

An activity view of repositories or people you watch. An organization's News Feed shows activity on repositories owned by the organization.

non-fast-forward

When your local copy of a repository is out-of-sync with the upstream repository and you need to fetch the upstream changes before you push your local changes.

notification

Updates, delivered by either the web or email depending on your settings, that give you information about the activities you're interested in.

OAuth app

A third-party application that uses access tokens rather than passwords to access information for users.

OAuth token

The access token used by OAuth apps to access information for users.

open source

Open source software is software that can be freely used, modified, and shared (in both modified and unmodified form) by anyone. Today the concept of "open source" is often extended beyond software, to represent a philosophy of collaboration in which working materials are made available online for anyone to fork, modify, discuss, and contribute to.

organization

Organizations are a group of two or more users that typically mirror real-world organizations. They are administered by users and can contain both repositories and teams.

organization owner

Users who have full administrative access to the organization they own.

origin

The default upstream repository. Most projects have at least one upstream project that they track. By default, origin is used for that purpose.

outside collaborator

A user who has been given access to one or more of an organization's repositories, but has no other access to the organization and is not a member of the organization.

owner

Organization members that have complete administrative access to the organization.

parent team

Within nested teams, the main team from which child teams inherit access permissions and @mentions.

participating notifications

A notification about an update in a conversation in an issue or pull request where your username or team was mentioned or where you have previously replied in a comment.

permalink

A permanent static hyperlink to a particular web page.

personal account

A GitHub account that belongs to an individual user.

pinned repository

A repository that a user has decided to display prominently on their profile.

pre-receive hooks

Scripts that run on the GitHub Enterprise server that you can use to implement quality checks.

primary email address

The main email address where GitHub sends receipts, credit card or PayPal charges, and other billing-related communication.

private contributions

Contributions made to a private (vs. public) repository.

private repository

Private repositories are only visible to the repository owner and collaborators that the owner specified.

production branch

A branch with final changes that are ready to be used or deployed to an application or site.

profile

The page that shows information about a user's activity on GitHub.

profile photo

A custom image users upload to GitHub to identify their activity, usually along with their username. This is also referred to as an avatar.

project board

Boards within GitHub that are made up of issues, pull requests, and notes that are categorized as cards in columns.

protected branch

Protected branches block several features of Git on a branch that a repository administrator chooses to protect. They can't be force pushed, deleted, have changes merged without required checks passing or required reviews approved, or have files uploaded to it from the GitHub web interface. A protected branch is usually the default branch.

public contributions

Contributions made to a public (vs. private) repository.

public repository

A public repository can be viewed by anyone, including people who aren't GitHub users.

pull

Pull refers to when you are fetching in changes and merging them. For instance, if someone has edited the remote file you're both working on, you'll want to pull in those changes to your local copy so that it's up to date. See also ["fetch"](#).

pull access

A synonym for read access.

pull request

Pull requests are proposed changes to a repository submitted by a user and accepted or rejected by a repository's collaborators. Like issues, pull requests each have their own discussion forum.

pull request review

Comments from collaborators on a pull request that approve the changes or request further changes before the pull request is merged.

pulse graph

A repository graph that gives you an overview of a repository's activity.

punch graph

A repository graph that shows the frequency of updates to a repository based on the day of week and time of day

push

To push means to send your committed changes to a remote repository on GitHub.com. For instance, if you change something locally, you can push those changes so that others may access them.

push a branch

When you successfully push a branch to a remote repository, you update the remote branch with changes from your local branch. When you "push a branch", Git will search for the branch's HEAD ref in the remote repository and verify that it is a direct ancestor to the branch's local HEAD ref. Once verified, Git pulls all objects (reachable from the local HEAD ref and missing from the remote repository) into the remote object database and then updates the remote HEAD ref. If the remote HEAD is not an ancestor to the local HEAD, the push fails.

push access

A synonym for write access.

read access

A permission level on a repository that allows the user to pull, or read, information from the repository. All public repositories give read access to all GitHub users. A synonym for pull access.

README

A text file containing information about the files in a repository that is typically the first file a visitor to your repository will see. A README file, along with a repository license, contribution guidelines, and a code of conduct, helps you share expectations and

manage contributions to your project.

rebase [↗](#)

To reapply a series of changes from a branch to a different base, and reset the HEAD of that branch to the result.

recovery code [↗](#)

A code that helps you regain access to your GitHub account.

release [↗](#)

GitHub's way of packaging and providing software to your users.

remote [↗](#)

This is the version of a repository or branch that is hosted on a server, most likely GitHub.com. Remote versions can be connected to local clones so that changes can be synced.

remote repository [↗](#)

A repository that is used to track the same project but resides somewhere else.

remote URL [↗](#)

The place where your code is stored: a repository on GitHub, another user's fork, or even a different server.

replica [↗](#)

A GitHub Enterprise instance that provides redundancy for the primary GitHub Enterprise instance.

repository [↗](#)

A repository is the most basic element of GitHub. They're easiest to imagine as a project's folder. A repository contains all of the project files (including documentation), and stores each file's revision history. Repositories can have multiple collaborators and can be either public or private.

repository cache [↗](#)

A read-only mirror of repositories for your GitHub Enterprise server instance, located near distributed teams and CI clients.

repository graph [↗](#)

A visual representation of your repository's data.

repository maintainer [↗](#)

Someone who manages a repository. This person may help triage issues and use labels and other features to manage the work of the repository. This person may also be responsible for keeping the README and contributing files updated.

required pull request review [↗](#)

Required reviews ensure that pull requests have at least one approved review before collaborators can make changes to a protected branch.

required status check [↗](#)

Checks on pull requests that ensure all required CI tests are passing before collaborators can make changes to a protected branch.

resolve [↗](#)

The action of fixing up manually what a failed automatic merge left behind.

revert [↗](#)

When you revert a pull request on GitHub, a new pull request is automatically opened, which has one commit that reverts the merge commit from the original merged pull request. In Git, you can revert commits with `git revert` .

review [↗](#)

Reviews allow others with access to your repository to comment on the changes proposed in pull requests, approve the changes, or request further changes before the pull request is merged.

root directory [↗](#)

The first directory in a hierarchy.

root filesystem [↗](#)

The base operating system and the GitHub Enterprise application environment.

saved reply [↗](#)

A comment you can save and add to your GitHub user account so that you can use it across GitHub in issues and pull requests.

scope [↗](#)

Named groups of permissions that an OAuth app or personal access token (classic) can request to access both public and non-public data.

seat

A user within a GitHub Enterprise organization. This may be referred to as "seat count."

secret team

A team that is only visible to the other people on the team and people with owner permissions.

security log

A log that lists the last 50 actions or those performed within the last 90 days.

server-to-server request

An API request used by an application that acts as a bot, independently of any particular user. For example, an application that runs on a scheduled basis and closes issues where there has been no activity for a long time. Applications that use this type of authentication don't use a licensed GitHub account so, in an enterprise with a billing plan that allows a certain number of licenses to be used, a server-to-server bot is not consuming one of your GitHub licenses. The token used in a server-to-server request is acquired programmatically, via the GitHub API. For more information, see "[Authenticating as a GitHub App installation](#)." See also, "[user-to-server request](#)."

service hook

Also called "webhook." Webhooks provide a way for notifications to be delivered to an external web server whenever certain actions occur on a repository or organization.

single sign-on

Also called SSO. Allows users to sign in to a single location - an identity provider (IdP) - that then gives the user access to other service providers.

snapshot

A checkpoint of a virtual machine at a point in time.

squash

To combine multiple commits into a single commit. Also a Git command.

SSH key

SSH keys are a way to identify yourself to an online server, using an encrypted message. It's as if your computer has its own unique password to another service. GitHub Enterprise Server uses SSH keys to securely transfer information to your computer.

staging instance

A way to test modifications before they are applied to your actual GitHub Enterprise instance.

star

A bookmark or display of appreciation for a repository. Stars are a manual way to rank the popularity of projects.

status

A visual representation within a pull request that your commits meet the conditions set for the repository you're contributing to.

status checks

Status checks are external processes, such as continuous integration builds, which run for each commit you make in a repository. For more information, see "[About status checks](#)."

subscription

A user or organization's GitHub plan.

team

A group of organization members that reflect your company or group's structure with cascading access permissions and mentions.

team maintainer

Organization members that have a subset of permissions available to organization owners to manage teams.

Team plan

An organization billing plan that gives you unlimited public and private repositories.

timeline

A series of events in a pull request or on a user profile.

topic branch

A regular Git branch that is used by a developer to identify a conceptual line of development. Since branches are very easy and inexpensive, it is often desirable to have several small branches that each contain very well defined concepts or small incremental yet related changes. Can also be called a feature branch.

topics

A way to explore repositories in a particular subject area, find projects to contribute to, and discover new solutions to a specific problem on GitHub.

traffic graph

A repository graph that shows a repository's traffic, including full clones (not fetches), visitors from the past 14 days, referring sites, and popular content.

transfer

To transfer a repository means to change the owner of a repository. The new owner will be able to immediately administer the repository's contents, issues, pull requests, releases, and settings.

upstream

When talking about a branch or a fork, the primary branch on the original repository is often referred to as the "upstream", since that is the main place that other changes will come in from. The branch/fork you are working on is then called the "downstream". Also called origin.

upstream branch

The default branch that is merged into the branch in question (or the branch in question is rebased onto). It is configured via `branch.<name>.remote` and `branch.<name>.merge`. If the upstream branch of A is origin/B sometimes we say "A is tracking origin/B".

user

Users are people with personal GitHub accounts. Each user has a personal profile, and can own multiple repositories, public or private. They can create or be invited to join organizations or collaborate on another user's repository.

user-to-server request

An API request used by an application that performs a task on behalf of a particular user. Where a task is carried out with user-to-server authentication it's shown on GitHub as having been done by a user via an application. For example, you might choose to create an issue from within a third-party application, and the application would do this on your behalf on GitHub. The scope of tasks an application can perform using a user-to-server request is restricted by both the app's and the user's permissions and access. The token used in a user-to-server request is acquired via OAuth. For more information, see "[Authenticating with a GitHub App on behalf of a user](#)." See also, "[server-to-server request](#)."

username

A user's handle on GitHub.

visible team

A team that can be viewed and @mentioned by every organization member.

watch

You can watch a repository or issue to receive notifications when updates are made to an issue or pull request.

watching notifications

A notification about activity in a repository a user has subscribed to.

web notifications

Notifications displayed in the web interface on GitHub: <https://github.com/notifications>

webhooks

Webhooks allow you to build or set up GitHub Apps which subscribe to certain events on GitHub.com. Webhooks provide a way for notifications to be delivered to an external web server whenever certain actions occur on a repository or organization. Also called a service hook.

write access

A permission level on a repository that allows the user to push, or write, changes to the repository.

Further reading

- [The Official Git Glossary](#)
- [Git documentation](#)
- [Git command list](#)

Legal

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