About authentication to GitHub - GitHub Docs

Authorizing for SAML single sign-on ⋮ 10-12 minutes

You can securely access your account's resources by authenticating to GitHub, using different credentials depending on where you authenticate.

About authentication to GitHub

To keep your account secure, you must authenticate before you can access certain resources on GitHub. When you authenticate to GitHub, you supply or confirm credentials that are unique to you to prove that you are exactly who you declare to be.

You can access your resources in GitHub in a variety of ways: in the browser, via GitHub Desktop or another desktop application, with the API, or via the command line. Each way of accessing GitHub supports different modes of authentication.

- Username and password with two-factor authentication, or a passkey
- Personal access token
- SSH key

Authenticating in your browser

If you're a member of an enterprise with managed users, you will authenticate to GitHub in your browser using your IdP. For more information, see "About Enterprise Managed Users" in the GitHub Enterprise Cloud documentation.

If you're not a member of an enterprise with managed users, you will authenticate using your GitHub.com username and password, or a passkey. You may also use two-factor authentication and SAML single sign-on, which can be required by organization and enterprise owners.

Note: As of March 2023, GitHub required all users who contribute code on GitHub.com to enable one or more forms of two-factor authentication (2FA). If you were in an eligible group, you would have received a notification email when that group was selected for enrollment, marking the beginning of a 45-day 2FA enrollment period, and you would have seen banners asking you to enroll in 2FA on GitHub.com. If you didn't receive a notification, then you were not part of a group required to enable 2FA, though we strongly recommend it.

For more information about the 2FA enrollment rollout, see this blog post.

If you need to use multiple accounts on GitHub.com, such as a personal account and a service account, you can quickly switch between your accounts without always needing to reauthenticate each time. For more information, see "Switching between accounts."

- **Username and password only**
    - You'll create a password when you create your account on GitHub. We recommend that you use a password manager to generate a random and unique password. For more information, see "Creating a strong password."
    - If you have not enabled 2FA, GitHub may ask for additional verification when you first sign in from a new or unrecognized device, such as a new browser profile, a browser where the cookies have been deleted, or a new computer. For more information, see "Verifying new devices when signing in."
- **Two-factor authentication (2FA)** (recommended)
    - If you enable 2FA, after you successfully enter your username and password, we'll also prompt you to provide a code that's generated by a time-based one time password (TOTP) application on your mobile device or sent as a text message (SMS).
    - After you configure 2FA, your account enters a check up period for 28 days. You can leave the check up period by successfully performing 2FA within those 28 days. If you don't perform 2FA in that timespan, you'll then be asked to perform 2FA inside one of your existing GitHub.com sessions.
    - If you cannot perform 2FA to pass the 28th day checkup, you will be provided a shortcut that lets you reconfigure your 2FA settings. You must reconfigure your settings before you can access the rest of GitHub.

For more information, see "Accessing GitHub using two-factor authentication" and "Configuring two-factor authentication."

  - In addition to authentication with a TOTP application or a text message, you can optionally add an alternative method of authentication with GitHub Mobile or a security key using WebAuthn. For more information, see "Configuring two-factor authentication" and "Configuring two-factor authentication."

    Note: If you cannot use any recovery methods, you have permanently lost access to your account. However, you can unlink an email address tied to the locked account. The unlinked email address can then be linked to a new or existing account. For more information, see "Unlinking your email address from a locked account."

- **Passkey**

  - You can add a passkey to your account to enable a secure, passwordless login. Passkeys satisfy both password and 2FA requirements, so you can complete your sign in with a single step. See "About passkeys."

- **SAML single sign-on**

  - Before you can access resources owned by an organization or enterprise account that uses SAML single sign-on, you may need to also authenticate through an IdP. For more information, see "About authentication with SAML single sign-on" in the GitHub Enterprise Cloud documentation.

Session cookies

GitHub uses cookies to provide services and secure GitHub.com. You can review details about GitHub's cookies in the privacy/cookies repository.

- The gist.github.com and github.com domains use separate cookies.
- GitHub typically marks a user session for deletion after two weeks of inactivity.
- GitHub does not immediately delete a session when you sign out. Periodically, GitHub automatically deletes expired sessions.

Authenticating with GitHub Desktop

You can authenticate with GitHub Desktop using your browser. For more information, see "Authenticating to GitHub in GitHub Desktop."

Authenticating with the API

You can authenticate with the API in different ways. For more information, see "Authenticating to the REST API."

Authenticating to the API with a personal access token

If you want to use the GitHub REST API for personal use, you can create a personal access token. If possible, GitHub recommends that you use a fine-grained personal access token instead of a personal access token (classic). For more information about creating a personal access token, see "Managing your personal access tokens."

Authenticating to the API with an app

If you want to use the API on behalf of an organization or another user, GitHub recommends that you use a GitHub App. For more information, see "About authentication with a GitHub App."

You can also create an OAuth token with an OAuth app to access the REST API. However, GitHub recommends that you use a GitHub App instead. GitHub Apps allow more control over the access and permission that the app has.

Authenticating to the API in a GitHub Actions workflow

If you want to use the API in a GitHub Actions workflow, GitHub recommends that you authenticate with the built-in `GITHUB_TOKEN` instead of creating a token. You can grant permissions to the `GITHUB_TOKEN` with the `permissions` key.

Note that `GITHUB_TOKEN` can only access resources within the repository that contains the workflow. If you need to make changes to resources outside of the workflow repository, you will need to use a personal access token or GitHub App.

For more information, see "Automatic token authentication."

Authenticating with the command line

You can access repositories on GitHub from the command line in two ways, HTTPS and SSH, and both have a different way of authenticating. The method of authenticating is determined based on whether you choose an HTTPS or SSH remote URL when you clone the repository. For more information about which way to access, see "About remote repositories."

HTTPS

You can work with all repositories on GitHub over HTTPS, even if you are behind a firewall or proxy.

If you authenticate with GitHub CLI, you can either authenticate with a personal access token or via the web browser. For more information about authenticating with GitHub CLI, see `gh auth login`.

If you authenticate without GitHub CLI, you must authenticate with a personal access token. When Git prompts you for your password, enter your personal access token. Alternatively, you can use a credential helper like Git Credential Manager. Password-based authentication for Git has been removed in favor of more secure authentication methods. For more information, see "Managing your personal access tokens." Every time you use Git to authenticate with GitHub, you'll be prompted to enter your credentials to authenticate with GitHub, unless you cache them with a credential helper.

SSH

You can work with all repositories on GitHub over SSH, although firewalls and proxies might refuse to allow SSH connections.

If you authenticate with GitHub CLI, the CLI will find SSH public keys on your machine and will prompt you to select one for upload. If GitHub CLI does not find a SSH public key for upload, it can generate a new SSH public/private keypair and upload the public key to your account on GitHub.com. Then, you can either authenticate with a personal access token or via the web browser. For more information about authenticating with GitHub CLI, see `gh auth login`.

If you authenticate without GitHub CLI, you will need to generate an SSH public/private keypair on your local machine and add the public key to your account on GitHub.com. For more information, see "Generating a new SSH key and adding it to the ssh-agent." Every time you use Git to authenticate with GitHub, you'll be prompted to enter your SSH key passphrase, unless you've stored the key.

To use a personal access token or SSH key to access resources owned by an organization that uses SAML single sign-on, you must also authorize the personal token or SSH key. For more information, see "Authorizing a personal access token for use with SAML single sign-on" or "Authorizing an SSH key for use with SAML single sign-on" in the GitHub Enterprise Cloud documentation.

GitHub's token formats

GitHub issues tokens that begin with a prefix to indicate the token's type.

Token type	Prefix	More information
Personal access token (classic)	`ghp_`	"Managing your personal access tokens"
Fine-grained personal access token	`github_pat_`	"Managing your personal access tokens"
OAuth access token	`gho_`	"Authorizing OAuth apps"
User access token for a GitHub App	`ghu_`	"Authenticating with a GitHub App on behalf of a user"
Installation access token for a GitHub App	`ghs_`	"Authenticating as a GitHub App installation"
Refresh token for a GitHub App	`ghr_`	"Refreshing user access tokens"