Troubleshooting custom domains and GitHub Pages

In this article

CNAME errors

DNS misconfiguration

Custom domain names that are unsupported

HTTPS errors

URL formatting on Linux

Browser cache

Domain name taken

You can check for common errors to resolve issues with custom domains or HTTPS for your GitHub Pages site.

GitHub Pages is available in public repositories with GitHub Free and GitHub Free for organizations, and in public and private repositories with GitHub Pro, GitHub Team, GitHub Enterprise Cloud, and GitHub Enterprise Server. For more information, see "[GitHub’s plans]."

CNAME errors

If you are publishing from a custom GitHub Actions workflow, any CNAME file is ignored and is not required.

If you are publishing from a branch, custom domains are stored in a CNAME file in the root of your publishing source. You can add or update this file through your repository settings or manually. For more information, see "[Managing a custom domain for your GitHub Pages site]."

For your site to render at the correct domain, make sure your CNAME file still exists in the repository. For example, many static site generators force push to your repository, which can overwrite the CNAME file that was added to your repository when you configured your custom domain. If you build your site locally and push generated files to GitHub Enterprise Cloud, make sure to pull the commit that added the CNAME file to your local repository first, so the file will be included in the build.

Then, make sure the CNAME file is formatted correctly.

- The CNAME filename must be all uppercase.
- The CNAME file can contain only one domain. To point multiple domains to your site, you must set up a redirect through your DNS provider.
- The CNAME file must contain the domain name only. For example, `www.example.com`, `blog.example.com`, or `example.com`.
- The domain name must be unique across all GitHub Pages sites. For example, if another repository's CNAME file contains `example.com`, you cannot use `example.com` in the CNAME file for your repository.

DNS misconfiguration

If you have trouble pointing the default domain for your site to your custom domain, contact your DNS provider.

You can also use one of the following methods to test whether your custom domain's DNS records are configured correctly:

- A CLI tool such as `dig`. For more information, see "Managing a custom domain for your GitHub Pages site".
- An online DNS lookup tool.

Custom domain names that are unsupported

If your custom domain is unsupported, you may need to change your domain to a supported domain. You can also contact your DNS provider to see if they offer forwarding services for domain names.

Make sure your site does not:

- Use more than one apex domain. For example, both `example.com` and `anotherexample.com`.
- Use more than one `www` subdomain. For example, both `www.example.com` and `www.anotherexample.com`.
- Use both an apex domain and custom subdomain. For example, both `example.com` and `docs.example.com`.

  The one exception is the `www` subdomain. If configured correctly, the `www` subdomain is automatically redirected to the apex domain. For more information, see "Managing a custom domain for your GitHub Pages site."

> **Warning:** We strongly recommend that you do not use wildcard DNS records, such as `*.example.com`. These records put you at an immediate risk of domain takeovers, even if you verify the domain. For example, if you verify `example.com` this prevents someone from using `a.example.com` but they could still take over `b.a.example.com` (which is covered by the wildcard DNS record). For more information, see "Verifying your custom domain for GitHub Pages."

For a list of supported custom domains, see "About custom domains and GitHub Pages."

HTTPS errors

GitHub Pages sites using custom domains that are correctly configured with `CNAME`, `ALIAS`, `ANAME`, or `A` DNS records can be accessed over HTTPS. For more information, see "Securing your GitHub Pages site with HTTPS."

It can take up to an hour for your site to become available over HTTPS after you configure your custom domain. After you update existing DNS settings, you may need to remove and re-add your custom domain to your site's repository to trigger the process of enabling HTTPS. For more information, see "Managing a custom domain for your GitHub Pages site."

If you're using Certification Authority Authorization (CAA) records, at least one CAA record must exist with the value `letsencrypt.org` for your site to be accessible over HTTPS. For more information, see "Certificate Authority Authorization (CAA)" in the Let's Encrypt documentation.

URL formatting on Linux

If the URL for your site contains a username or organization name that begins or ends with a dash, or contains consecutive dashes, people browsing with Linux will receive a server error when they attempt to visit your site. To fix this, change your GitHub Enterprise Cloud username to remove non-alphanumeric characters. For more information, see "Changing your GitHub username."

Browser cache

If you've recently changed or removed your custom domain and can't access the new URL in your browser, you may need to clear your browser's cache to reach the new URL. For more information on clearing your cache, see your browser's documentation.

Domain name taken

If you're trying to use a custom domain and it says the domain is already in use, you can make the domain available for your own use by verifying it first. For more information, see "Verifying your custom domain for GitHub Pages."