GitHub Docs

Fork a repo

In this article

A fork is a new repository that shares code and visibility settings with the original "upstream" repository.

Mac   Windows   Linux

GitHub CLI   Desktop   Web browser

About forks 🔗

A fork is a new repository that shares code and visibility settings with the original "upstream" repository. Forks are often used to iterate on ideas or changes before they are proposed back to the upstream repository, such as in open source projects or when a user does not have write access to the upstream repository. For more information, see "[Working with forks](#)."

Propose changes to someone else's project 🔗

For example, you can use forks to propose changes related to fixing a bug. Rather than logging an issue for a bug you have found, you can:

- Fork the repository.
- Make the fix.
- Submit a pull request to the project owner.

Use someone else's project as a starting point for your own idea. 🔗

Open source software is based on the idea that by sharing code, we can make better, more reliable software. For more information, see the "[About the Open Source Initiative](#)" on the Open Source Initiative.

For more information about applying open source principles to your organization's development work on your GitHub Enterprise Server instance, see GitHub's white paper "[An introduction to innersource](#)."

When creating your public repository from a fork of someone's project, make sure to include a license file that determines how you want your project to be shared with others. For more information, see "[Choose an open source license](#)" at

choosealicense.com.

For more information on open source, specifically how to create and grow an open source project, we've created [Open Source Guides](#) that will help you foster a healthy open source community by recommending best practices for creating and maintaining repositories for your open source project. You can also take a free [GitHub Skills](#) course on maintaining open source communities.

Prerequisites 🔗

If you haven't yet, first set up Git and authentication with your GitHub Enterprise Server instance from Git. For more information, see "[Set up Git](#)."

Forking a repository 🔗

You might fork a project to propose changes to the upstream repository. In this case, it's good practice to regularly sync your fork with the upstream repository. To do this, you'll need to use Git on the command line. You can practice setting the upstream repository using the same [octocat/Spoon-Knife](#) repository you just forked.

1. On your GitHub Enterprise Server instance, navigate to the [octocat/Spoon-Knife](#) repository.

2. In the top-right corner of the page, click **Fork**.

3. Under "Owner," select the dropdown menu and click an owner for the forked repository.

4. By default, forks are named the same as their upstream repositories. Optionally, to further distinguish your fork, in the "Repository name" field, type a name.

5. Optionally, in the "Description" field, type a description of your fork.

6. Optionally, select **Copy the DEFAULT branch only**.

   For many forking scenarios, such as contributing to open-source projects, you only need to copy the default branch. If you do not select this option, all branches will be copied into the new fork.

7. Click **Create fork**.

> **Note:** If you want to copy additional branches from the upstream repository, you can do so from the **Branches** page. For more information, see "[Creating and deleting branches within your repository](#)."

To create a fork of a repository, use the `gh repo fork` subcommand.

```
gh repo fork REPOSITORY
```

To create the fork in an organization, use the `--org` flag.

```
gh repo fork REPOSITORY --org "octo-org"
```

You can fork a repository on GitHub.com or in GitHub Desktop. For information about forking on GitHub.com, see the web browser version of this article.

In GitHub Desktop, if you attempt to clone a repository that you don't have write access to, a fork is automatically created for you.
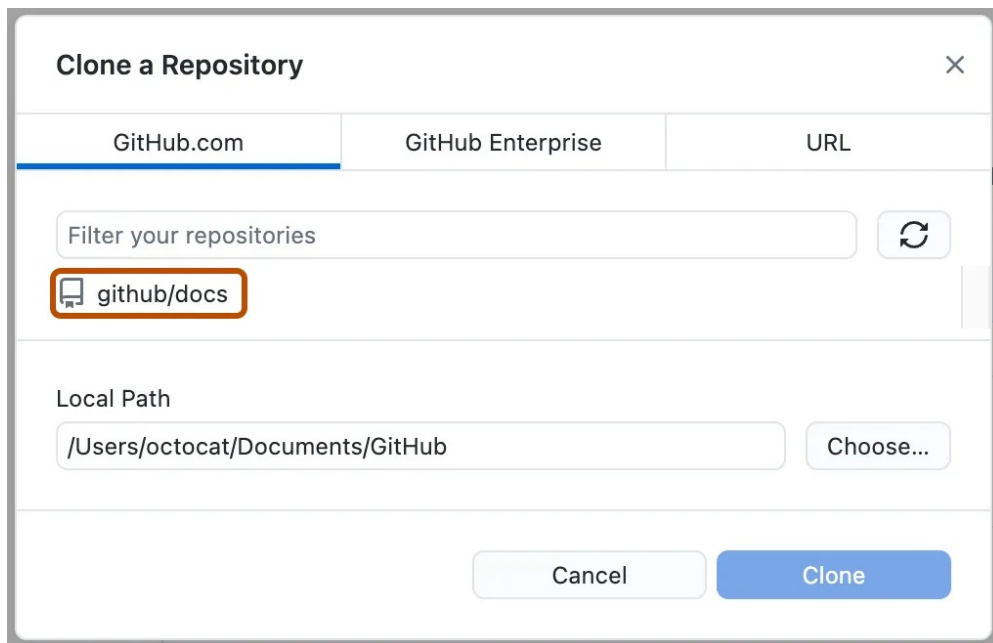
1. In the **File** menu, click **Clone Repository**.

2. Click the tab that corresponds to the location of the repository you want to clone. You can also click **URL** to manually enter the repository location.
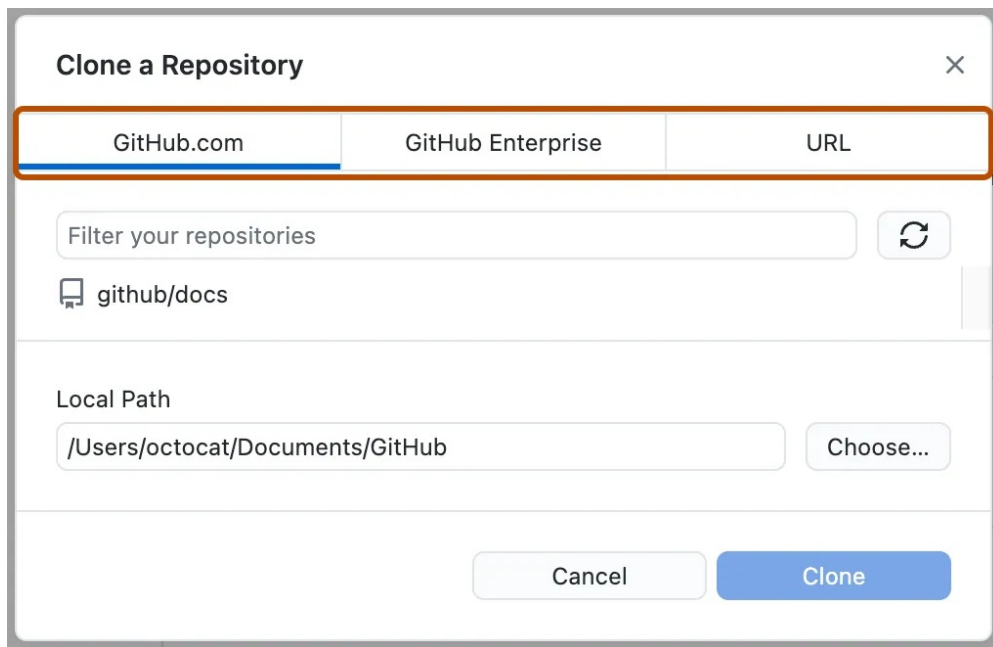
3 From the list of repositories, click the repository you want to clone.

4 To select the local directory into which you want to clone the repository, next to the "Local Path" field, click **Choose...** and navigate to the directory.
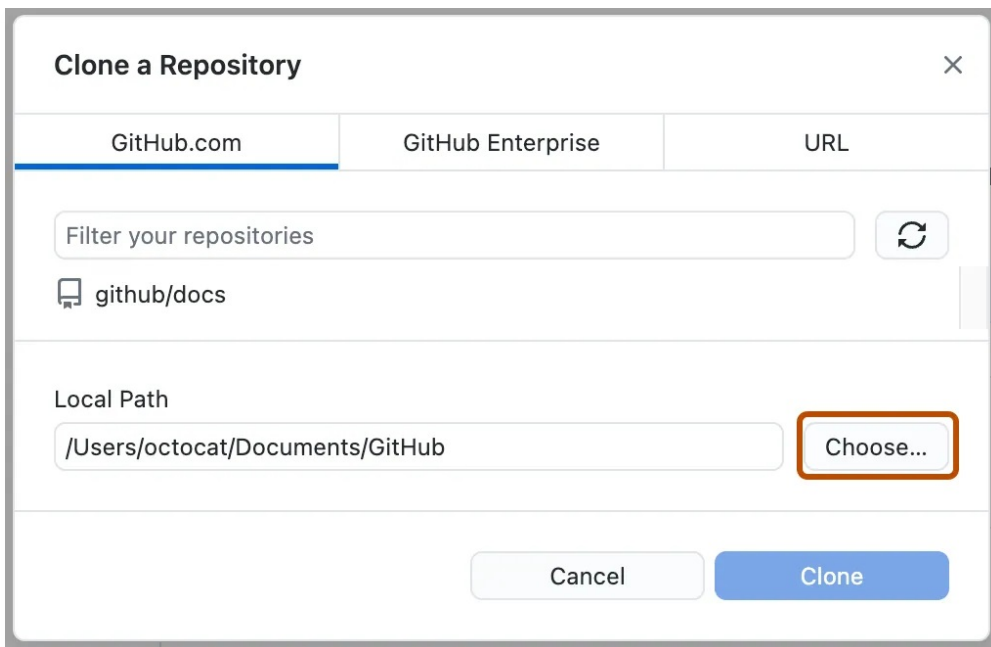
5 At the bottom of the "Clone a Repository" window, click **Clone**.

6 Read the information in the "How are you planning to use this fork?" window.

   ○ If you plan to use this fork for contributing to the original upstream repository,
     click **To contribute to the parent project**.

   ○ If you plan to use this fork for a project not connected to the upstream, click **For
     my own purposes**.

7 Click **Continue**.

Cloning your forked repository 🔗

Right now, you have a fork of the Spoon-Knife repository, but you do not have the files in
that repository locally on your computer.

1 On your GitHub Enterprise Server instance, navigate to **your fork** of the Spoon-
   Knife repository.
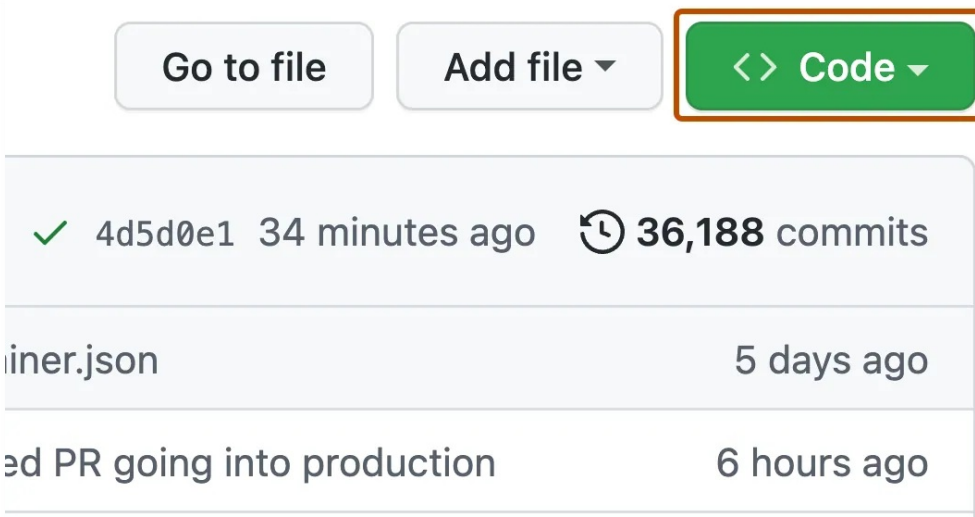
2 Above the list of files, click ‹› **Code**.

3 Copy the URL for the repository.

- To clone the repository using HTTPS, under "HTTPS", click ⧉.
- To clone the repository using an SSH key, including a certificate issued by your organization's SSH certificate authority, click **SSH**, then click ⧉.
- To clone a repository using GitHub CLI, click **GitHub CLI**, then click ⧉.

| Go to file | Add file ▾ | <> Code ▾ |

| Local | Codespaces |

y.

>_ **Clone**   ⓘ

or

n   **HTTPS**   SSH   GitHub CLI

ry   https://github.com/github/docs.git   ⧉

3   Use Git or checkout with SVN using the web URL.

ag   ⊡ **Open with GitHub Desktop**

te

7   ⊟ **Download ZIP**

4 Open TerminalTerminalGit Bash.

5 Change the current working directory to the location where you want the cloned directory.

6 Type `git clone`, and then paste the URL you copied earlier. It will look like this, with your GitHub Enterprise Server username instead of `YOUR-USERNAME`:

```
git clone https://HOSTNAME/YOUR-USERNAME/Spoon-Knife
```

7 Press **Enter**. Your local clone will be created.

```
$ git clone https://HOSTNAME/YOUR-USERNAME/Spoon-Knife
> Cloning into `Spoon-Knife`...
> remote: Counting objects: 10, done.
> remote: Compressing objects: 100% (8/8), done.
> remote: Total 10 (delta 1), reused 10 (delta 1)
> Unpacking objects: 100% (10/10), done.
```
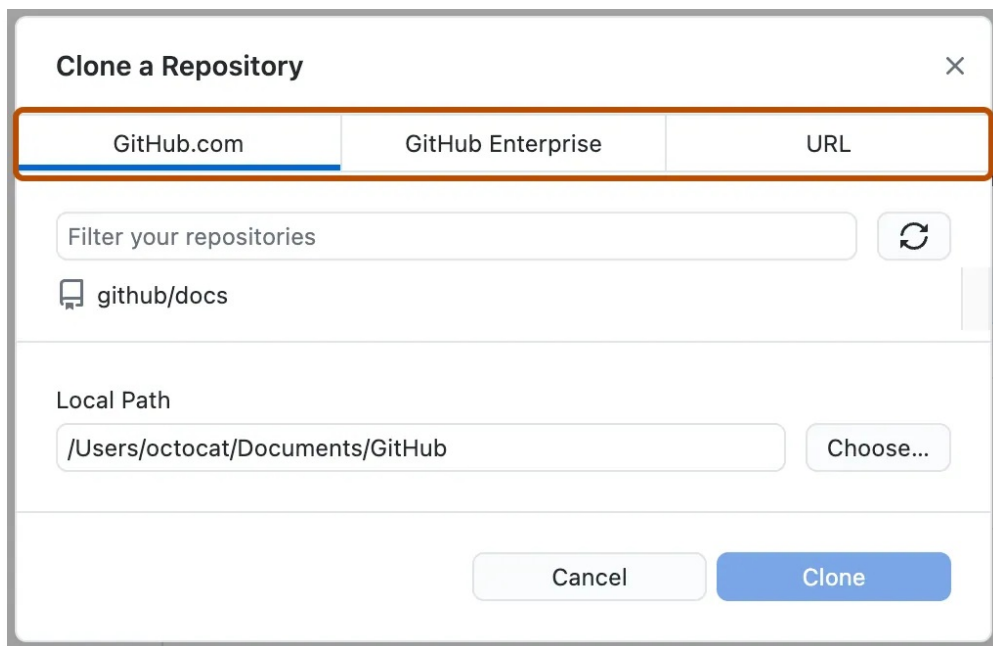
To learn more about GitHub CLI, see "About GitHub CLI."

To create a clone of your fork, use the `--clone` flag.

```
gh repo fork REPOSITORY --clone=true
```

1 In the **File** menu, click **Clone Repository**.
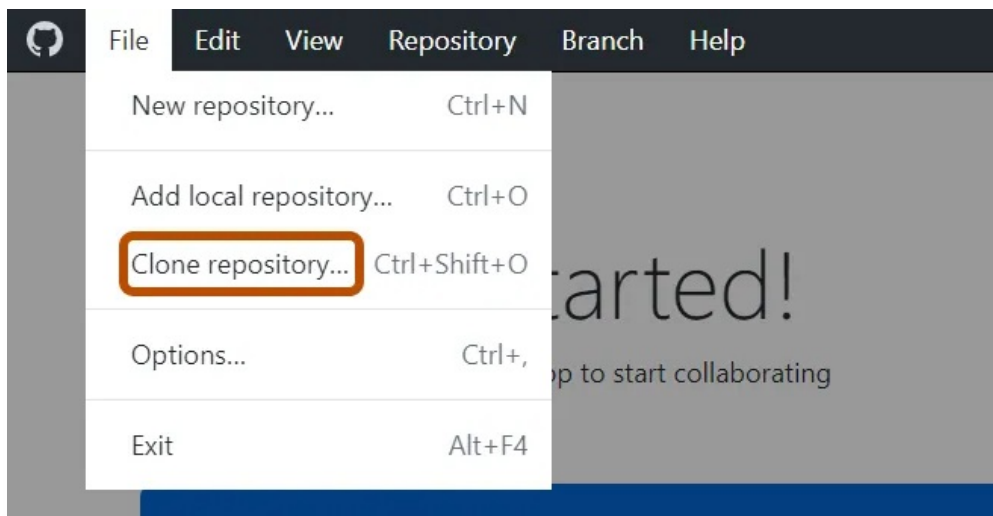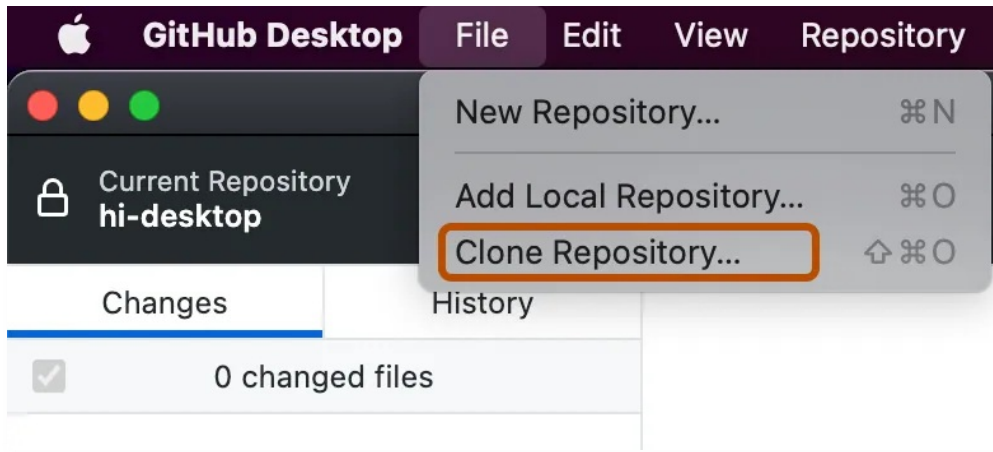
2 Click the tab that corresponds to the location of the repository you want to clone. You can also click **URL** to manually enter the repository location.

3 From the list of repositories, click the repository you want to clone.

4. To select the local directory into which you want to clone the repository, next to the "Local Path" field, click **Choose...** and navigate to the directory.

5. At the bottom of the "Clone a Repository" window, click **Clone**.

Configuring Git to sync your fork with the upstream repository 🔗

When you fork a project in order to propose changes to the upstream repository, you can configure Git to pull changes from the upstream repository into the local clone of your fork.

1. On your GitHub Enterprise Server instance, navigate to the octocat/Spoon-Knife repository.

2. Above the list of files, click ‹› **Code**.

3. Copy the URL for the repository.

   - To clone the repository using HTTPS, under "HTTPS", click 📋.
   - To clone the repository using an SSH key, including a certificate issued by your organization's SSH certificate authority, click **SSH**, then click 📋.
   - To clone a repository using GitHub CLI, click **GitHub CLI**, then click 📋.

4. Open TerminalTerminalGit Bash.

5. Change directories to the location of the fork you cloned.

   - To go to your home directory, type just `cd` with no other text.
   - To list the files and folders in your current directory, type `ls` .
   - To go into one of your listed directories, type `cd your_listed_directory` .
   - To go up one directory, type `cd ..` .

6. Type `git remote -v` and press **Enter**. You will see the current configured remote

repository for your fork.

```
$ git remote -v
> origin  https://HOSTNAME/YOUR_USERNAME/YOUR_FORK.git (fetch)
> origin  https://HOSTNAME/YOUR_USERNAME/YOUR_FORK.git (push)
```

7 Type `git remote add upstream`, and then paste the URL you copied in Step 3 and press **Enter**. It will look like this:

```
git remote add upstream https://HOSTNAME/ORIGINAL_OWNER/Spoon-Knife.git
```

8 To verify the new upstream repository you have specified for your fork, type `git remote -v` again. You should see the URL for your fork as `origin`, and the URL for the upstream repository as `upstream`.

```
$ git remote -v
> origin    https://HOSTNAME/YOUR_USERNAME/YOUR_FORK.git (fetch)
> origin    https://HOSTNAME/YOUR_USERNAME/YOUR_FORK.git (push)
> upstream  https://HOSTNAME/ORIGINAL_OWNER/ORIGINAL_REPOSITORY.git (fetch)
> upstream  https://HOSTNAME/ORIGINAL_OWNER/ORIGINAL_REPOSITORY.git (push)
```

Now, you can keep your fork synced with the upstream repository with a few Git commands. For more information, see "Syncing a fork."

> To learn more about GitHub CLI, see "About GitHub CLI."

To configure a remote repository for the forked repository, use the `--remote` flag.

```
gh repo fork REPOSITORY --remote=true
```

To specify the remote repository's name, use the `--remote-name` flag.

```
gh repo fork REPOSITORY --remote-name "main-remote-repo"
```

Editing a fork 🔗

You can make any changes to a fork, including:

- **Creating branches:** *Branches* allow you to build new features or test out ideas without putting your main project at risk.
- **Opening pull requests:** If you want to contribute back to the upstream repository, you can send a request to the original author to pull your fork into their repository by submitting a pull request.

Find another repository to fork 🔗

Fork a repository to start contributing to a project. You can fork a private or internal repository to your personal account or to an organization on your GitHub Enterprise Server instance where you have permission to create repositories, provided that the settings for the repository and your enterprise policies allow forking. Generally, you can fork any public repository to your personal account or to an organization where you have permission to create repositories.

For more information about when you can fork a repository, see "[About permissions and visibility of forks](#)."

Next Steps 🔗

You have now forked a repository, practiced cloning your fork, and configured an upstream repository.

- For more information about cloning the fork and syncing the changes in a forked repository from your computer, see "[Set up Git](#)."

- You can also create a new repository where you can put all your projects and share the code on GitHub. Creating a repository for your project allows you to store code in GitHub. This provides a backup of your work that you can choose to share with other developers. For more information, see "[Create a repository](#).""

- Each repository on GitHub is owned by a person or an organization. You can interact with the people, repositories, and organizations by connecting and following them on GitHub Enterprise Server. For more information, see "[Be social](#)."

- GitHub has a great support community where you can ask for help and talk to people from around the world. Join the conversation on [GitHub Community](#).