Fork A Repo

MAC | WINDOWS | LINUX

A *fork* is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project.

Most commonly, forks are used to either propose changes to someone else's project or to use someone else's project as a starting point for your own idea.

Propose changes to someone else's project

A great example of using forks to propose changes is for bug fixes. Rather than logging an issue for a bug you've found, you can:

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

If the project owner likes your work, they might pull your fix into the original repository!

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

At the heart of open source is the idea that by sharing code, we can make better, more reliable software.

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 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.

Fork an example repository

Forking a repository is a simple two-step process. We've created a repository for you to practice with!

1. On your GitHub Enterprise instance, navigate to the octocat/Spoon-Knife repository.
2. In the top-right corner of the page, click **Fork**.

That's it! Now, you have a *fork* of the original octocat/Spoon-Knife repository.

Keep your fork synced

You might fork a project in order to propose changes to the *upstream*, or original, 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!

Article versions

GitHub.com
GitHub Enterprise 2.11
GitHub Enterprise 2.10
GitHub Enterprise 2.9
GitHub Enterprise 2.8

Step 1: Set Up Git

If you haven't yet, you should first set up Git. Don't forget to set up authentication to your GitHub Enterprise instance from Git as well.

Step 2: Create a local clone of your fork

Right now, you have a fork of the Spoon-Knife repository, but you don't have the files in that repository on your computer. Let's create a *clone* of your fork locally on your computer.

1. On GitHub Enterprise, navigate to **your fork** of the Spoon-Knife repository.
2. Under the repository name, click **Clone or download**.

3. In the Clone with HTTPs section, click to copy the clone URL for the repository.

4. Open Terminal.
5. Type `git clone`, and then paste the URL you copied in Step 2. It will look like this, with your GitHub Enterprise username instead of `YOUR-USERNAME`:

```
$ git clone https://hostname/YOUR-USERNAME/Spoon-Knife
```

6. 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.
remove: Total 10 (delta 1), reused 10 (delta 1)
Unpacking objects: 100% (10/10), done.
```

Now, you have a local copy of your fork of the Spoon-Knife repository!

Step 3: Configure Git to sync your fork with the original Spoon-Knife repository

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

1. On GitHub Enterprise, navigate to the octocat/Spoon-Knife repository.
2. Under the repository name, click **Clone or download**.

3. In the Clone with HTTPs section, click to copy the clone URL for the repository.

4. Open Terminal.
5. Change directories to the location of the fork you cloned in Step 2: Create a local clone of your fork.
   - 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'll 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 2 and press **Enter**. It will look like this:

```
$ git remote add upstream https://hostname/octocat/Spoon-Knife.git
```

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

Next Steps

The sky's the limit with the changes you can make 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 are hoping to contribute back to the original 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 any public repository, or any private repository you can access. For more information, see "About forks."

Celebrate

You have now forked a repository, practiced cloning your fork, and configured an upstream repository. What do you want to do next?

- "Set up Git"
- "Create a repository"
- **Fork A Repository**
- "Be social"

Contact a human