Importing an external Git repository using the command line

If your source code is tracked in a Git repository, you can import the repository using Git on the command line.

Before you start, make sure you know:

- Your GitHub Enterprise Server username
- The clone URL for the external repository, such as `https://external-host.com/user/repo.git` or `git://external-host.com/user/repo.git` (perhaps with a `user@` in front of the `external-host.com` domain name)

> For purposes of demonstration, we'll use:
>
> - An external account named **extuser**
> - An external Git host named `https://external-host.com`
> - A GitHub Enterprise Server personal account named **ghuser**
> - A repository on your GitHub Enterprise Server instance named **repo.git**

1. [Create a new repository on GitHub Enterprise Server](). You'll import your external Git repository to this new repository.

2. On the command line, make a "bare" clone of the external repository using the external clone URL. This creates a full copy of the data, but without a working directory for editing files, and ensures a clean, fresh export of all the old data.

```
$ git clone --bare https://external-host.com/EXTUSER/REPO.git
# Makes a bare clone of the external repository in a local directory
```

3. Push the locally cloned repository to GitHub Enterprise Server using the "mirror" option, which ensures that all references, such as branches and tags, are copied to the imported repository.

```
$ cd REPO.git
$ git push --mirror https://HOSTNAME/USER/REPO.git
# Pushes the mirror to the new repository on your GitHub Enterprise Server
instance
```

4. Remove the temporary local repository.

```
cd ..
rm -rf REPO.git
```

If the repository you are importing contains large files, you may run into a warning or error. For more information on large files and how to manage them, see "[About large

[files on GitHub](#)."