



Creating a GitHub Repository

Jeffrey Leek, Assistant Professor of Biostatistics
Johns Hopkins Bloomberg School of Public Health

Recap: Git vs. GitHub

- You don't need GitHub to use Git
- Git = Local (on your computer); GitHub = Remote (on the web)
- GitHub allows you to:
 1. Share your repositories with others
 2. Access other users' repositories
 3. Store remote copies of your repositories (on GitHub's server) in case something happens to your local copies (on your computer)

Creating a GitHub Repository

- Two methods of creating a GitHub repository:
 1. Start a repository from scratch
 2. "Fork" another user's repository
- We'll start with the first method
- *NOTE: A repository is often referred to as a "repo"*

Start a Repository From Scratch

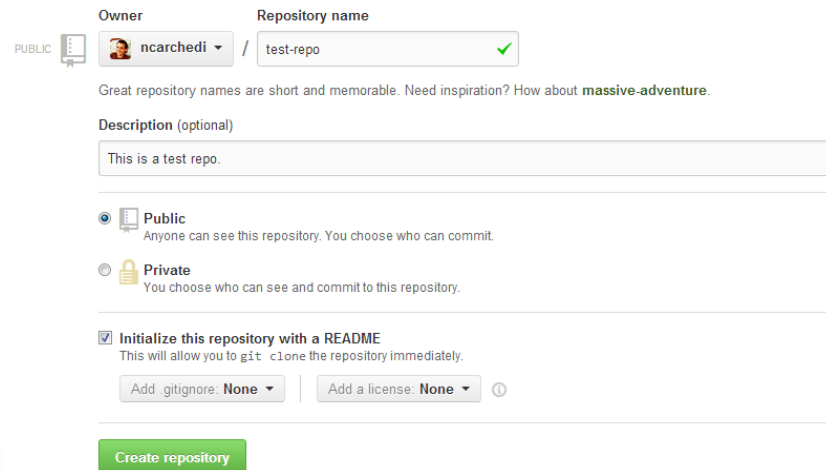
- Either go to your profile page (<https://github.com/yourUserNameHere/>) and click on "Create a new repo" in the upper righthand corner of the page

...OR...


- Go directly to <https://github.com/new> (you'll need to log into your GitHub account if you haven't already done so)

Start a Repository From Scratch

- Create a name for your repo and type a brief description of it
- Select "Public" (Private repos require a paid [or education] account)
- Check the box next to "Initialize this repository with a README"
- Click the "Create repository" button



The screenshot shows the GitHub 'Create new repository' form. At the top, there are two input fields: 'Owner' with a dropdown menu showing 'ncarchedi' and a 'PUBLIC' icon, and 'Repository name' with the text 'test-repo' and a green checkmark. Below these fields is a text input for 'Description (optional)' containing 'This is a test repo.'. Under the 'Visibility' section, the 'Public' option is selected with a radio button, and the 'Private' option is unselected. Below this, the 'Initialize this repository with a README' checkbox is checked. At the bottom, there are two dropdown menus for 'Add .gitignore: None' and 'Add a license: None', followed by a green 'Create repository' button.

Owner:  ncarchedi / Repository name: test-repo ✓

Great repository names are short and memorable. Need inspiration? How about [massive-adventure](#).

Description (optional): This is a test repo.

☒ **Public**
Anyone can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

☒ **Initialize this repository with a README**
This will allow you to `git clone` the repository immediately.

Add .gitignore: **None** | Add a license: **None** ⓘ

Create repository

Start a Repository From Scratch

- Congratulations! You've created a GitHub repository.

The screenshot shows the GitHub interface for a newly created repository. At the top, it says 'PUBLIC' and 'ncarchedi / test-repo'. To the right are buttons for 'Unwatch' (1), 'Star' (0), and 'Fork' (0). Below this, a message says 'This is a test repo. — Edit'. A summary bar shows '1 commit', '1 branch', '0 releases', and '1 contributor'. Below the summary, there's a green 'Initial commit' button and a dropdown for 'branch: master'. The repository name 'test-repo' is followed by a plus icon. The main content area shows the 'Initial commit' by 'ncarchedi' with the message 'latest commit bceef8fc7d'. Below this, there's a 'README.md' file. The file content shows the repository name 'test-repo' and the message 'This is a test repo.'.

PUBLIC ncarchedi / test-repo

Unwatch 1 Star 0 Fork 0

This is a test repo. — Edit

1 commit 1 branch 0 releases 1 contributor

branch: master test-repo / +

Initial commit

ncarchedi authored in a few seconds latest commit bceef8fc7d

README.md Initial commit in a few seconds

README.md

test-repo

This is a test repo.

<> Code

Issues 0

Pull Requests 0

Wiki

Pulse

Graphs

Network

Settings

HTTPS clone URL

You can clone with [HTTPS](#), [SSH](#), or [Subversion](#).

Creating a Local Copy

- Now you need to create a copy of this repo on your computer so that you can make changes to it
- Open Git Bash
- Create a directory on your computer where you will store your copy of the repo:

```
$ mkdir ~/test-repo
```

- Navigate to this new directory using the following command:

```
$ cd ~/test-repo
```

Creating a Local Copy

- Initialize a local Git repository in this directory

```
$ git init
```

- Point your local repository at the remote repository you just created on the GitHub server

```
$ git remote add origin https://github.com/yourUserNameHere/test-repo.git
```


Creating a Local Copy

- Here's what this process looks like in action:

```
Welcome to Git (version 1.8.4-preview20130916)

Run 'git help git' to display the help index.
Run 'git help <command>' to display help for specific commands.

Nick@NICK-PC ~
$ mkdir ~/test-repo

Nick@NICK-PC ~
$ cd ~/test-repo

Nick@NICK-PC ~/test-repo
$ git init
Initialized empty Git repository in c:/Users/Nick/test-repo/.git/

Nick@NICK-PC ~/test-repo (master)
$ git remote add origin https://github.com/ncarchedi/test-repo.git

Nick@NICK-PC ~/test-repo (master)
$ _
```

Fork a Another User's Repository

- The second method of creating a repository is to make a copy of someone else's
- This process is called "forking" and is an important aspect of open-source software development
- Begin by navigating to the desired repository on the GitHub website and click the "Fork" button shown below



<https://help.github.com/articles/fork-a-repo>

Clone the Repo

- You now have a copy of the desired repository on your GitHub account
- Need to make a local copy of the repo on your computer
- This process is called "cloning" and can be done using the following command:

```
$ git clone https://github.com/yourUserNameHere/repoNameHere.git
```

- *NOTE: This will clone the repository into your current directory.*

What Else?

- If you make changes to your local copy of the repo, you'll probably want to push your changes to GitHub at some point
- You also may be interested in staying current with any changes made to the original repository from which you forked your copy
- We will cover some more Git/GitHub basics in coming lectures, but in the meantime, here are some great resources:
 - <https://help.github.com/articles/fork-a-repo>
 - <http://git-scm.com/book/en/Git-Basics-Getting-a-Git-Repository>