‹ (../03-branching/)

Introduction to Git and GitHub (../)

› (../05-remote-collaboration/)

Getting started with GitHub

Overview

Teaching: 25 min
Exercises: 0 min
Questions

- What is a remote repository?
- How can I use GitHub to work from multiple locations?

Objectives

- Understand how to set up remote repository
- Understand how to push local changes to a remote repository
- Understand how to clone a remote repository

We're going to set up a remote repository that we can use from multiple locations. The remote repository can also be shared with colleagues, if we want to.

GitHub

GitHub (http://GitHub.com) is a company which provides remote repositories for Git and a range of functionalities supporting their use. GitHub allows users to set up their private and public source code Git repositories. It provides tools for browsing, collaborating on and documenting code. GitHub, like other services such as GitLab (https://about.gitlab.com/) and Bitbucket (https://bitbucket.org), supports a wealth of resources to support projects including:

- Browsing code from within a web browser, with syntax highlighting
- Software release management
- Issue and bug tracking
- Project management tools

GitHub for research

GitHub **isn't** the only remote repositories provider. It is however very popular, in particular within the Open Source communities. The reason why we teach GitHub in this tutorial is mainly due to popular demand.

Also, GitHub has functionality which is particularly useful for researchers (https://github.com/blog/1840-improving-github-for-sciences) such as making code citable!

Get an account

Let’s get back to our tutorial. We will first need a GitHub account.

Sign up (https://GitHub.com) or if you already have an account sign in (https://GitHub.com).

Create a new repository

Now, we can create a repository on GitHub,

- Log in to GitHub (https://GitHub.com/)
- Click on the **Create** icon on the top right
- Enter Repository name: “papers”
- For the purpose of this exercise we’ll create a public repository
- Since we’ll be importing a local repository, make sure that **Initialize this repository with a README** is **unselected**
- Click **Create Repository**

You’ll get a page with new information about your repository. We already have our local repository and we will be *pushing* it to GitHub, so this is the option we will use:

```
$ git remote add origin https://github.com/<USERNAME>/papers.git
$ git push -u origin master
```

The first line sets up an alias `origin`, to correspond to the URL of our new repository on GitHub.

Push locally tracked files to a remote repository

Now copy and paste the second line,

```
$ git push -u origin master
```

```
Counting objects: 32, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (28/28), done.
Writing objects: 100% (32/32), 3.29 KiB | 0 bytes/s, done.
Total 32 (delta 7), reused 0 (delta 0)
To https://github.com/gcapes/papers.git
 * [new branch]      master -> master
Branch master set up to track remote branch master from origin.
```

This **pushes** our `master` branch to the remote repository, named via the alias `origin` and creates a new `master` branch in the remote repository.

Now, on GitHub, we should see our code and if we click the `Commits` tab we should see our complete history of commits.

Our local repository is now available on GitHub. So, anywhere we can access GitHub, we can access our repository!

Push other local branches to a remote repository

Let’s push each of our local branches into our remote repository:

```
$ git push origin branch_name
```

The branch should now be created in our GitHub repository.

To list all branches (local and remote):

```
$ git branch -a
```

Deleting branches (for information only)

Don’t do this now. This is just for information. To delete branches, use the following syntax:

```
$ git branch -d <branch_name>                  # For local branches
$ git push origin --delete <branch_name>        # For remote branches
```

Cloning a remote repository

Now, let’s do something drastic! (but before that step, **make sure that you pushed all your local branches into the remote repository**)

```
$ cd ..
$ rm -rf papers
```

Gulp! We’ve just wiped our local repository! But, as we’ve a copy on GitHub we can just copy, or `git clone` that,

```
$ git clone https://github.com/<USERNAME>/papers.git
```

```
Cloning into 'papers'...
remote: Counting objects: 32, done.
remote: Compressing objects: 100% (21/21), done.
remote: Total 32 (delta 7), reused 32 (delta 7), pack-reused 0
Unpacking objects: 100% (32/32), done.
Checking connectivity... done.
```

Cloning creates an exact copy of the repository. By default it creates a directory with the same name as the name of the repository.

Now, if we change into *papers* we can see that we have our repository,

```
$ cd papers
$ git log
```

and we can see our Git configuration files too:

```
$ ls -A
```

In order to see the other branches locally, we can check them out as before:

```
$ git branch -r                                  # Show remote branches
$ git checkout paperWJohn                        # Check out the paperWJohn branch
```

Push changes to a remote repository

We can use our cloned repository just as if it was a local repository so let’s make some changes to our files and commit these.

```
$ git checkout master                            # We'll continue working on the master branch
$ nano journal.md                                # Add results section
$ git add journal.md                             # Stage changes
$ git commit
```

Having done that, how do we send our changes back to the remote repository? We can do this by *pushing* our changes,

```
$ git push origin master
```

If we now check our GitHub page we should be able to see our new changes under the *Commit* tab.

To see all remote repositories (we can have multiple!) type:

```
$ git remote -v
```

Key Points

- Git is the version control system: GitHub is a remote repositories provider.
- `git clone` to make a local copy of a remote repository
- `git push` to send local changes to remote repository

‹ (../03-branching/)

› (../05-remote-collaboration/)

Edit on GitHub (https://github.com/emdupre/git-course/edit/gh-pages/_episodes/04-remote.md) / Contributing (https://github.com/emdupre/git-course/blob/gh-pages/CONTRIBUTING.md) / Source (https://github.com/emdupre/git-course/) / Cite (https://github.com/emdupre/git-course/blob/gh-pages/CITATION) / Contact (mailto:elizabeth.dupre@mail.mcgill.ca?subject=Git course)