Git Walk-through

Contents

[What is Git? 2](#_Toc25135083)

[What is GitHub? 2](#_Toc25135084)

[Who uses GitHub? 2](#_Toc25135085)

[Why should I use GitHub? 3](#_Toc25135086)

[What is a repository? 4](#_Toc25135087)

[How Do I Get a GitHub Account? 4](#_Toc25135088)

[Setting up two-factor authentication 5](#_Toc25135089)

[Recovery codes. 6](#_Toc25135090)

[Git Workflow 6](#_Toc25135091)

[What is a pull request ? 6](#_Toc25135092)

[Pre-pull Requests 7](#_Toc25135093)

[Squash 7](#_Toc25135094)

[Stash/Pop 7](#_Toc25135095)

[Tagging 7](#_Toc25135096)

[Issues 8](#_Toc25135097)

# What is Git?

Git is an open source project started by Linux creator Linus Torvalds.

It’s an example of a DVCS (hence Distributed Version Control System). Rather than have only one single place for the full version history of the software as is common in once-popular version control systems like CVS or Subversion, in Git, every developer's working copy of the code is also a repository that can contain the full history of all changes.

Although it’s mostly used for code, Git can be used to manage any other type of file. Think of it as a filing system for every draft of a document

# What is GitHub?

GitHub is a Git repository hosting service, but it adds many of its own features. While Git is a command line tool, GitHub provides a Web-based graphical interface. It also provides access control and several collaboration features, such as a wikis and basic task management tools for every project.

## Who uses GitHub?

Pretty much everybody.

* Google
* Facebook
* Microsoft
* Twitter
* Paypal
* Yahoo
* Linkedin
* Dropbox

# Why should I use GitHub?

Have you ever?

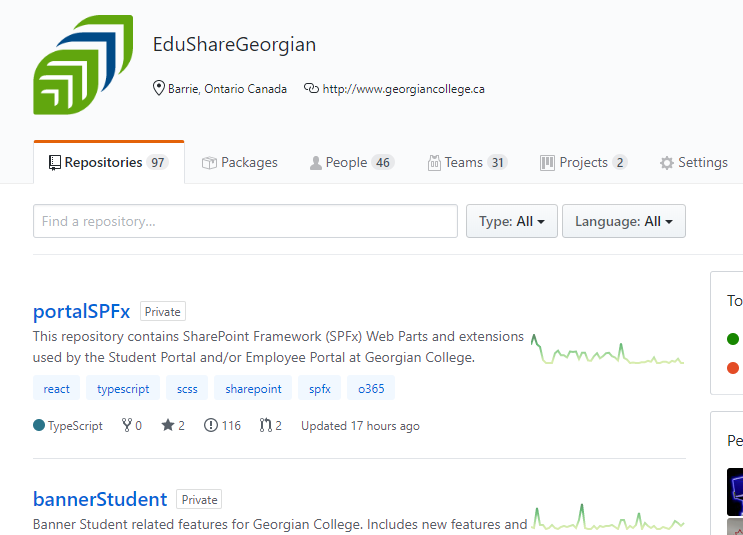
* Made a change to code, realized it was a mistake and wanted to revert back?
* Lost code or had a backup that was too old?
* Had to maintain multiple versions of a product?
* Wanted to see the difference between two (or more) versions of your code?
* Wanted to prove that a particular change broke or fixed a piece of code?
* Wanted to review the history of some code?
* Wanted to submit a change to someone else's code?
* Wanted to share your code, or let other people work on your code?
* Wanted to see how much work is being done, and where, when and by whom?
* Wanted to experiment with a new feature without interfering with working code?

"Code doesn’t exist unless it’s checked into a version control system. Use version control for everything you do. Master it and use it."

**The code repository is the source of truth when it comes to production code. All production updates MUST be sourced from your repository.**

# What is a repository?

A repository is the most basic element of Git. It is easiest to imagine as a project's folder. However, unlike an ordinary folder on your laptop, a GitHub repository offers simple yet powerful tools for collaborating with others. A repository contains all of the project files (including documentation), and stores each file's revision history.

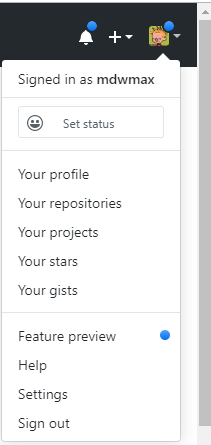


# How Do I Get a GitHub Account?

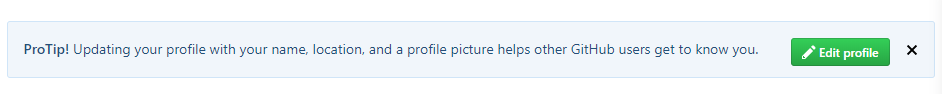
Go to [github.Com](https://github.com/EduShareGeorgian/portal/wiki/License-Requirements) and follow the instructions.

## Setting up two-factor authentication

In the upper right corner click on your avatar and select Your Profile.



Click on the Edit Profile Button



### Adding your SMS number

Click the Security link on the left. Scroll down to the two-factor authentication section. Under the Two-factor methods select SMS number and click Add. Follow the instructions to add your mobile phone number.

### Recovery codes.

Recovery codes can be used to access your account in the event you lose access to your device and cannot receive two-factor authentication codes.

Click the Security link on the left. Scroll down to the two-factor authentication section. Under the Two-factor methods select Recovery codes and click Add. Follow the instructions create your recovery codes.

Treat your recovery codes with the same level of attention as you would your password! Save them with a password manager.

# Georgian on GitHub

We've created an organization account (EduShareGeorgian) for Georgian on GitHub. Our current plan is Platinum ($200 usd per month) which give us the following:

* Unlimited Members
* Unlimited Public Repositories (any one can see these)
* 125 Private Repositories

# How do I access the Georgian GitHub Repositories?

You need to be invited. Once you've created an account, send your username to one of (Todd Hiles or Mike Westbrooke) and they will invite you to the Georgian GitHub site.

# How to Clone a Repository

Cloning a repository means that you're downloading a copy of the source code from source control.

Command line: git clone <https://github.com/EduShareGeorgian/training.git>

GitIgnore

gitignore tells git which files (or patterns) it should ignore. It's usually used to avoid committing transient files from your working directory that aren't useful to other collaborators, such as compilation products, temporary files IDEs create, etc.

## Git Workflow

Your local repository consists of three "trees" maintained by git. The first one is your Working Directory which holds the actual files. The second one is the Index which acts as a staging area and finally the HEAD which points to the last commit you've made.



## What is a pull request ?

Pull requests let you tell others about changes you've pushed to a GitHub repository. Once a pull request is sent, interested parties can review the set of changes, discuss potential modifications, and even push follow-up commits if necessary.

## Pre-pull Requests

## Squash

With git it’s possible to squash previous commits into one. This is a great way to group certain changes together before sharing them with others. ~ Here’s how to squash some commits into one. Let’s say this is your current git log.

## Stash/Pop

Stashing takes the dirty state of your working directory — that is, your modified tracked files and staged changes — and saves it on a stack of unfinished changes that you can reapply at any time.

## Tagging

A tag represents a version of a particular branch at a moment in time. Usually you'll tag a particular version so that you can recreate it, e.g., this is the sprint 22 release.

## Issues

To close an issue in the same repository, use one of the keywords in the list below followed by a reference to the issue number in the commit message. For example, a commit message with Fixes #45 will close issue 45 in that repository once the commit is merged into the default branch.

* close
* closes
* closed
* fix
* fixes
* fixed
* resolve
* resolves
* resolved