

Learning Outcomes

After completing this lab, a student will be able to

1. **Log in** to the *Gitea* server on `cs-devel.potsdam.edu` with a new password they picked.
2. Find the **S24-405** course organization where class source code will be distributed.
3. **Clone** a specific repository from the server using `git` on a CS lab computer.
4. Create organizations, teams, and repositories in `Git`.
5. Do simple management of a `Git` repository.

Deliverables

There are two (2) deliverables for this assignment:

1. `README` file in an acceptable plain-text format (`.md`, `.txt`, `.org`)
2. `HelloWorld.java`

Submission is through the CS Department Gitea.

Introduction

Overview

`Git` is an essential tool for developing software as part of a team. Even if you have used `Git` as an individual developer, in this course you will learn more complex skills, such as resolving merge conflicts, that support source code versioning with multiple developers.

If you have used `Git` and the CS Gitea server before, you can skip to the next section. If you are new to `Git` or Gitea, continue reading “Square One”.

Square One

`git` is a *version control* program. You can think of it like a database of changes across some set of text files. It is designed to keep track of changes to a **set** of files over **time** and make sharing the versions of those files easy.

It is designed to handle any number of files, from the Java code used during a single lab section up to **all** of the Linux operating system kernel (so, tens of lines of code in a handful of files up to tens of *millions* of lines of code across thousands of files). In this class you will use `git` (a program running on your local machine) to connect to a server running on `cs-devel.potsdam.edu` to download *sample code*, helper code for *assignments*, and eventually as part of the team software project.

Using git

If you're new to the CS Department Gitea server, your account has been set up with a dummy password. You will log in to the server, find the *repository* (the *git* term for a set of files) associated with this assignment, copy the link for the *repo*, and download it locally.

1. (**New Gitea users**) Change your *password* on the *Gitea* server on `cs-devel.potsdam.edu`.

Point your browser at `https://cs-devel.potsdam.edu`. Your login name is the same as your Potsdam email address *without* `@potsdam.edu`. If you haven't used Gitea in the past, your password is `Ki10Ech0` (note the capitalization and use of digits). Because everyone with a new Gitea account has the same password, the first step is to change it.

Once you login, your Dashboard will be displayed. A randomly generated avatar image for you is shown in the upper-right of the screen. Clicking on it gives a drop down menu including **Settings**. Select **Settings**.

Your profile is shown with a horizontal menu across the top of then page. Find the **Account** entry, click on it, and the top of that page is a form to change your password. Change the password (it is **not** connected to your standard campus account so it will not change that password nor will changing the campus password change this one). Click the **Update Password** button when you have entered a password you will remember.

✓ Be sure that you can log *out* and then log back *in* to the *Gitea* server with your new password.

2. Your newly logged in to Dashboard is pretty empty but there is a pair of tabs near the center top, labeled **Repositories** and **Organizations**. Initially **Repositories** is selected. Click on **Organizations**.

You were added to the `S24-405` organization as an existing user or when your account was created. That organization is where the instructor will distribute source code. You are a member of the **Student** team in the organization, so you can read the contents but not change them. Click on the `S24-405` link.

You should now see the `p01-GiteaSetup` repo (along with any other repositories that have been posted to the class; this is where you are going to go to get source code that is distributed to the class and any code materials for assignments. Click on the `p01-GiteaSetup` link.

Finally, you are on the page for `S24-405/p01-GiteaSetup`. The repository shows the files and folders on this page. Just above the listing of the contents is a bright line. Below the right end of the line is a box with an URL in it, an URL beginning with `https://cs-devel.potsdam.edu` and so on. To the right of the text box is a copy icon; click on the copy icon to copy the URL to the clipboard. That URL is how the *git* program will find the repository to copy onto your local machine.

Open a terminal window on the local machine. Navigate to the folder where you want to download the repo (a new subfolder will be created with the repo contents). Assume I want `p01-GiteaSetup` copied into the `/CIS405` folder. In that folder I would type `git clone`, a space, and then I would paste the URL copied above. Like this:

```
~ $ cd CIS405
~/CIS405 $ git clone https://cs-devel.potsdam.edu/S24-405/p01-GiteaSetup.git
```

```
Cloning into 'p01-GiteaSetup' ...
Username for 'https://cs-devel.potsdam.edu': grabowlm
Password for 'https://grabowlm@cs-devel.potsdam.edu':
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.

~/CIS405 $
```

Hitting enter at `git clone` will prompt for your account name and password on the *Gitea* server and will then create a subfolder named `p01-GiteaSetup` below the current directory containing the contents of the repo. Note that the exact number of objects will differ depending on the repository being cloned.

✓ Ensure that you have successfully cloned the repo to your local machine.

Background for handin through git

The following Gitea help videos are available to you at any time:

- “Creating a Gitea Organization for Turnin”,
<http://cs.potsdam.edu/Classes/git-videos/CreatingAGiteaOrganizationForTurnIn.mp4>
- “Creating a local repo”,
<http://cs.potsdam.edu/Classes/git-videos/GitCreateLocalRepo.mp4>
- “Git repo turnin”,
<http://cs.potsdam.edu/Classes/git-videos/GitRepoTurnIn.mp4>
- “Clone course source code”,
<http://cs.potsdam.edu/Classes/git-videos/GiteaCloneCourseSourceCode.mp4>

Creating and managing a git organization and repository

3. The `git` version control system has been used to distribute class assignments and share source code with the class. This has been done through an *organization*, S24-405 on the **Gitea** server running on `cs-devel.potsdam.edu`.

The organization is a level of administration between the `git` repository (a single project/assignment) and the whole shared Departmental server (all the data on `cs-devel`). It can be used to limit access to a collection of repositories (“repos” from here on).

When you turn in your work using `git`, you *could* put it in a **public** repo on the server. A public repo can be *seen* and **cloned** by anyone with an account on `cs-devel`. This is *good* because your instructor can clone it to grade it; this is *bad* because unscrupulous students can clone it to turn it in as their own.

Creating an organization that *you* own means you can create and upload repos in the organization. Adding your instructor to the organization with **read-only** access means that they

can see and clone **private** repos owned by the organization. With only two members of the organization, only two accounts have access to the private repos.

- (a) Log in to your *Gitea* account on `cs-devel.potsdam.edu`.
 - (b) Create a new organization named `S24-405-<email>` where `<email>` is replaced with your campus email address (before the `@` sign). The **Organization** tab is on the dashboard (home) page, to the middle right. There is a '+' sign to add a new organization. Name your organization as above and make it *private*.
 - (c) Go to your new organization's dashboard page by clicking on its name beneath the **Organization** tab.
 - (d) Create a new *Instructor* team: beneath the list of teams, click **New Team**. Name the new team *Instructor* and let them access **All repositories**. You will see the access is *Read* which is fine. Finish creating your new team.
 - (e) Add your instructor to your team. On the team page, type the instructor's email address into the search bar, select them and click the **Add Team Member** button.
- ✓ Return to the *organization* dashboard and check that your organization has two teams and that the two teams that share access.
4. You will now create a *repository* owned by the *organization* you created in the previous checkpoint. The exact process is outlined in <http://cs.potsdam.edu/Classes/git-videos/GitRepoTurnIn.mp4>.

The video outline is:

* Set configuration values for git

Set the default branch name to main (instead of master) and set the user's email address. The `--global` means for all repositories under this account.

Remember to replace the "johndoe" with YOUR email address.

```
#+BEGIN_SRC bash
  git config --global user.name "John Doe"
  git config --global user.email johndoe@example.com
  git config --global init.defaultBranch main
#+END_SRC
```

* Create *local* repository in the directory where you're working

Empty `~git~` repository for assignment

```
#+BEGIN_SRC bash
  git init
#+END_SRC
```

* Edit files for assignment

Write a README file with your ID block in it and assignment information.

NOT IN THE VIDEO: Write `HelloWorld.java` to include in your repo. Keep it simple.

NOT IN THE VIDEO: Update your README file with `compile/run` instructions for `HelloWorld.j`

```

* Add changed files to local ~git~
#+BEGIN_SRC bash
    git add .
#+END_SRC

* Commit changes to local ~git~
#+BEGIN_SRC bash
    git commit
#+END_SRC
    Then fill out a commit message.

* Create a *remote* repository on ~cs-devel~
    Login
    Switch to your organization S24-405-johndoe
    Create *empty* repository p01-GiteaSetup
    Use instructions Gitea printed for us to connect
* Loop until done
** Make and commit more *local* changes
** Push *local* changes up to *remote* repo
    #+BEGIN_SRC bash
        git push origin main
    #+END_SRC

```

✓ Check the repository on the local machine and on `cs-devel`. To check the status of your local repo, run `git status` and/or `git log`. The `git status` command should show the following messages:

```

On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

```

The command `git log` will show a message similar to the following:

```

commit f2bd49d6195fdc61ed03c9bb9ccba8c47e98109c (HEAD -> main, origin/main)
Author: Laura Grabowski <grabowlm@potdam.edu>
Date: Sun Jan 21 16:04:25 2024 -0500

    init: README.md

```

The details will be replaced with the details of your repo (a different commit, your name, the date of the commit, and your commit message).

- Commit changes and push your updated files (`README` and `HelloWorld.java`) to your Gitea repo for this assignment.

✓ Check your Gitea repo to ensure that changes were committed.