Version Control Setup for class Gitlab access

Overview

You will need to install Git in order to share your work with your team. Your team project will be hosted in git.gccis.rit.edu. You will be **cloning** it once and then pushing and pulling from it constantly throughout the rest of this semester. Authenticating with just your username/password over HTTPS doesn't work directly with git anymore; you need either an **SSH key** or a **Personal Access Token (PAT)** instead. If you are feeling adventurous and/or know about SSH you can go for it but we will only be supporting and providing instructions for the faster **(PAT)** configuration.

If you're on **Windows**, Git for Windows ships with Git Credential Manager, which pops up a browser login the first time you push or pull and caches your credentials after that — so this is optional for you, but still more convenient once set up (no browser popups at all). If you're on **Linux or macOS**, you'll need to set up before you can clone anything.

Installing GIT and confirming access to GCCIS GitLab

Git is the version control tool you'll use to submit your project this semester — you'll be committing and pushing your work constantly, so this needs to be working before you touch any starter code.

- **Windows:** install Git for Windows. This also installs **Git Bash**, a terminal you'll use throughout the course — use it instead of Command Prompt or PowerShell for anything Git-related.
- **macOS:** it's strongly recommended that Mac users install Homebrew if you haven't already — it makes installing Git (and everything else this semester) a lot easier. Once you have it, run brew install git.

  Otherwise, you can install directly from git-scm.com.
- **Linux:** install via your distro's package manager, e.g. sudo apt install git on Ubuntu/Debian.

Verify it worked by opening a terminal (Git Bash on Windows) and running:

```
git --version
```

You should see a version number printed back, e.g. **git version 2.47.1**. or something like it

Get Access to GCCIS GitLab

The ***starter code*** that you will be given and the code you will be modifying will live on the GCCIS GitLab server, git.gccis.rit.edu. **You will not be able to access it until you've logged in at least once**, so don't put this off.

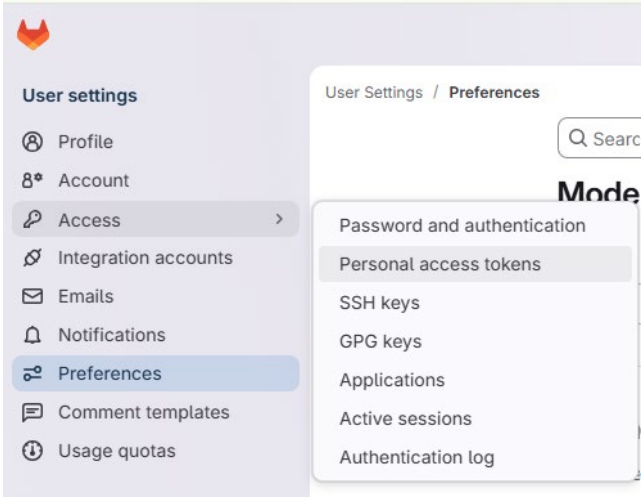
1. Go to git.gccis.rit.edu and sign in with your normal RIT username and password.
2. If this is your first time, you may be asked to authorize the connection between your RIT account and GitLab — accept it.

Logging in is enough to get started, but before you can actually clone, push, or pull a repository from the command line, you'll need either an SSH key or a Personal Access Token set up. Follow the steps below to accomplish that.

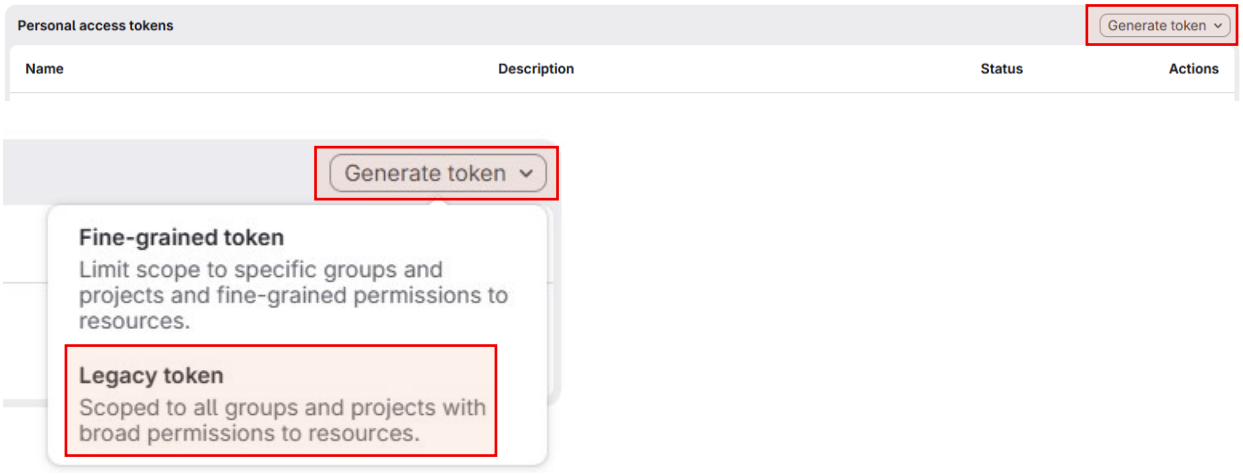
Personal Access Tokens

A **PAT** is a long random string you generate once and use as your password whenever GitLab asks for one over HTTPS — cloning, pushing, pulling, or any tool that needs API access.

1. In GitLab, click your **profile picture → Edit profile**, then in the left menu expand **Access → Personal access tokens**.



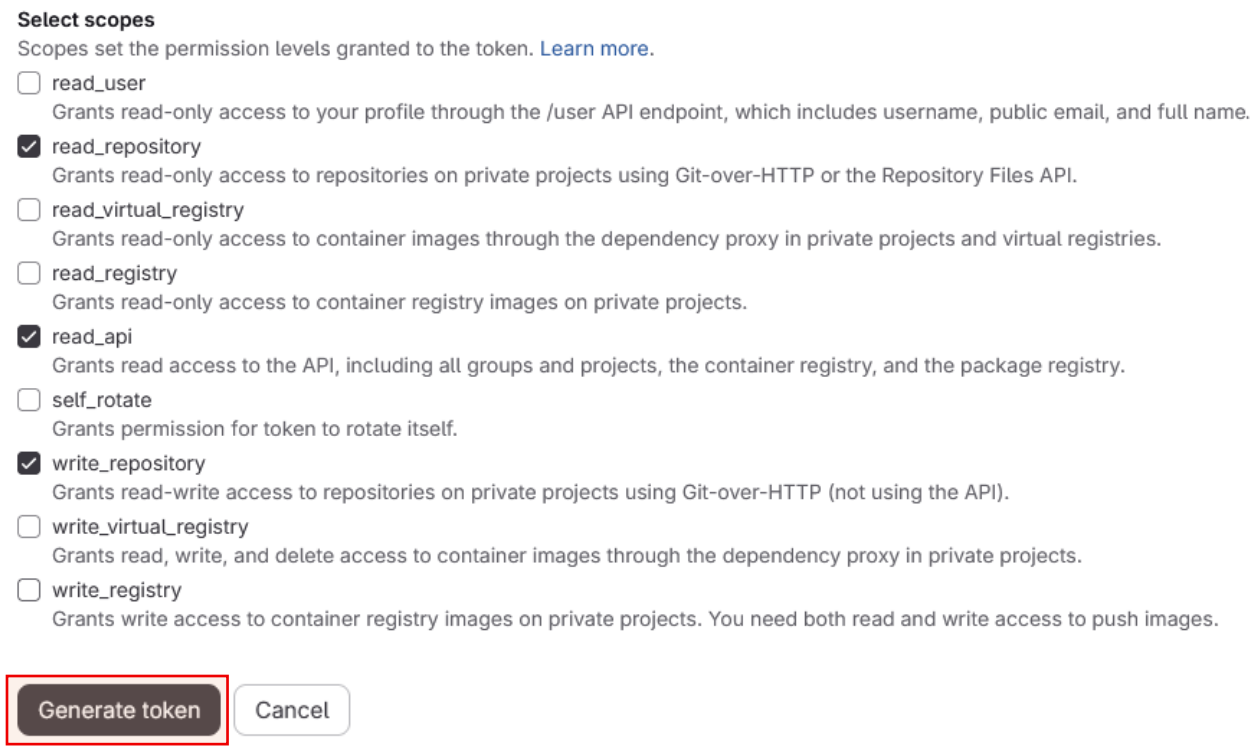
2. Click **Generate token** option and be sure to select **“Legacy token”**



3. You will need to fill in some required fields on the form.

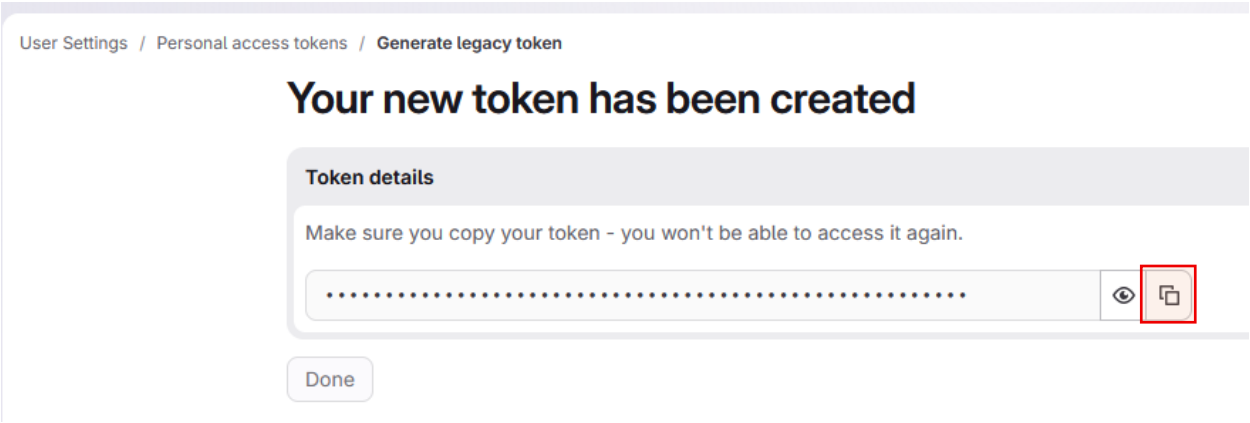
   - Give it a meaningful **Token name** (e.g. SWEN-261-team-project-token)
   - The **Expiration date** defaults to just one month out — change it to at least the end of the semester or push it **all the way to the maximum** allowed (about a year out; you'll see "An administrator has set the maximum expiration date to..." once you hit it). **Don't leave the one-month default**, or you'll be locked out mid-semester and have to do this again.

4. **Select scopes** - nothing is checked by default, so you'll need to check boxes yourself. Check at least **read_repository** , **write_repository** and **read_api**.



5. Click **Generate token**. **This is the only time you will ever be able to see or copy this token.** The moment you navigate away from this page, it's gone for good — GitLab stores it in a form that even GitLab itself might not show you again. If you lose it, your only option will likely be to revoke it and generate a new one from scratch.

   Copy it NOW and put it somewhere you'll actually be able to find it again — a text file in your Google Drive works well. Don't rely on remembering to paste it somewhere "*in a minute*."



   "Make sure you save it - you won't be able to access it again." This is your only chance.

6. When cloning with an HTTPS URL, use your GitLab username and paste the token **as the password** when prompted. Most git clients (and VS Code) will offer to remember it after the first time.

You should be all set for the next steps but if you haven't yet done so be sure to have VSCode Visual Studio Code installed in your development machine(s) as this is the editor we'll use in class and for project course materials.