Github Basics

Greg Chambers

2025-06-30

Table of contents

Github Overview

Github is a widely used tool for managing work and development on code-based projects. It has several key features:

1. Version control - Github is based on **git**, a version control software. Git allows you to track changes in a project over time, and revert to previous versions if problems arise. You can also create **branches** of a project to work on new/experimental features without breaking the **main** code which might need to remain functional with no downtime. After completing the branch, the changes can be merged into the main code.
2. Versatile - You can use Github to collaborate and track changes for any text- or script-based project. It is mostly used for hosting codes for web/software development or data analysis (SQL, SAS, R, Python), but you can also use it for complex documentation on sourdough bread making, data storage for public use, or reports on cyber-security threats.
3. Web hosted - Github is a web application that can host public and private **repositories** (or **repos**). Anyone can see and download information on a public repo. Private repos can only be seen by individuals with approved access. For our purposes, no sensitive code, PHI, passwords, tokens, or credentials can be stored in any repository, even if it is private. Github is great for working on projects for your own use, but it is also a fantastic way to share code with other users inside and outside TDH who might find it useful.
4. Free - Anyone can create a free Github account. To interact with TDH repos, you should create your own account using your tn.gov email and two-factor authentication. We can then add you to any TDH CEDEP teams in Github, so you can start contributing.

Organization

Our use of Github was only recently approved, so we are still working on the best way to integrate it into our teams. We have one main organization, TDH-CEDEP, with several program area subgroups (e.g., SSI, VH). Once you have your account, an admin can add you to the correct group(s). You will be able to view/download any public repository, but you can only see the private repositories for the teams you are a part of. You can also only make changes to repos for your teams.

If your program area does not have a team in TDH-CEDEP, let us know and we can create one easily. If you plan on having a lot of users and repos, it would be good to designate someone on your team to be an admin, so they can manage access permissions.

Security

The following steps should be closely followed in order to adhere to our agreed upon security protocol with STS:

1. Use your tn.gov email with two-factor authentication when setting up your Github account.
2. Admins must promptly remove access permissions from users who leave CEDEP or transition between teams. They should also perform regular user audits.
3. Only use Github for projects that need it. You can use Git locally, if you just need version control.
4. Keep repositories private unless the project needs to be viewable by other jurisdictions and/or the public.
5. Teams should develop and follow protocols to ensure that no sensitive data is published to any repos, public or private. See the security page for more details.

1 Security Details

STS took a lot of sweet talking before they approved our Github use, so we want to be on our best behavior (in addition to general data security concerns). As such, the following practices should be followed when using Github (expanded from the summary on the first page).

1.1 Account Setup

Use your tn.gov email to create a personal account (can we use our existing account and add our tn.gov email?). Enable two-factor authentication for increased security. You will be able to contribute to your own teams repos using your own account. [section on hosting your own repos on your own account]

1.2 Team Access

Admins for the TDH-CEDEP organization must maintain user access to the correct teams. This includes adding new members to the correct team, removing team members when they leave their positions, and performing regular user audits.

1.3 Do you even need Github?

If you just want version control on a project, you can just use Git locally. Github is great for sharing projects and working with outside parties, but if the process is internal, there might not be a point to hosting it on Github at all.

1.4 Public vs Private Repositories

The default choice for repositories should be to make them private. More public repos are just more ways for things to accidentally get published to the public. That being said, there are reasons why you might want a repo to be public. Public repos are great for sharing data and code with other parties, and even having them suggest improvements. You can even

publish data packages that are useful for TDH employees who might not have their own Github accounts.

It is important to note, that a repository being private should not affect the sort of information published there. PHI, passwords, etc. are not allowed in any repo, public or private. [should there be a middle tier of data that can be in private repos, but not public ones , e.g. network paths?]

1.5 Sensitive Data

The following data are not allowed to be published to any Github repository:

- PHI
- Non-public data (preliminary, data needing suppression, etc.)
- Usernames/Passwords/Credentials
- API tokens

In order to prevent this data from getting into Github, remove it from your codes and become used to excluding it from your work without consideration for if it will end up in Github. It is especially easy to leave notes or comments in code about test cases or outliers that might need to be deleted before saving a file.

You can also use gitignore functionality to hide certain files from Git, which can help when you need data for a code to run on your machine, but you don't want it in the version history that goes in Github.

[Consider a middle tier of info that can be in private but not public repos]

1.5.1 Removing Sensitive Data from a Repo

If sensitive data gets pushed to Github, we need to take it down as fast as possible. Follow the following steps:

1. Notify an admin
2. Make the repository private (if it isn't already)
3. Edit the repo to remove the sensitive data
4. Edit the version history to remove the sensitive data [develop guidance because this can be complex]. Note information needed for follow-up investigation (time published and removed, author, file source, etc.)

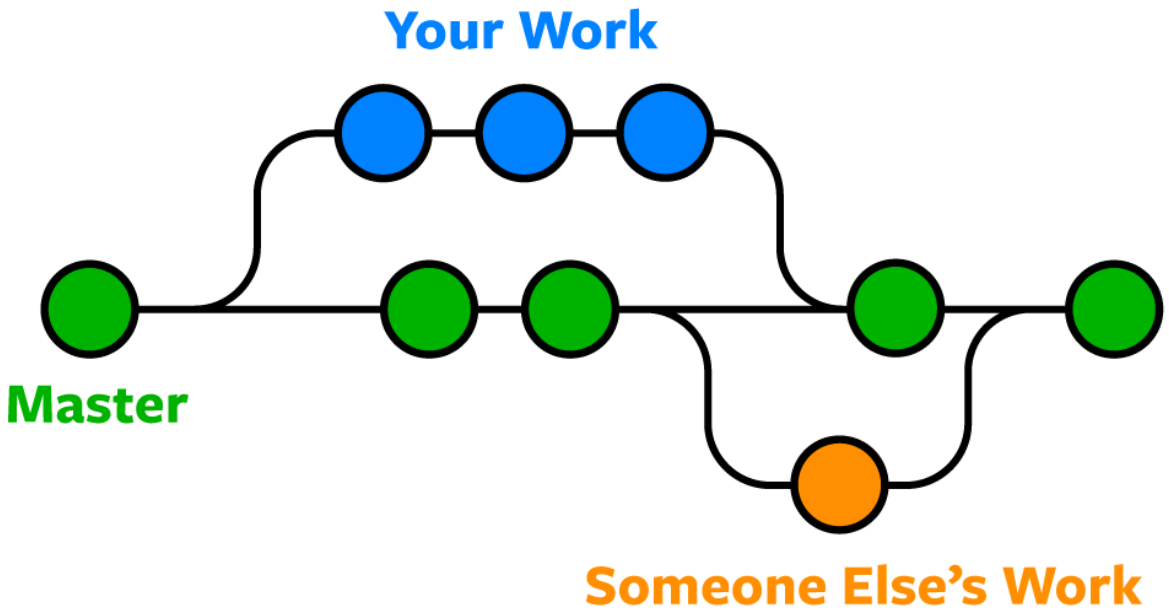
5. Notify anyone else who needs to know (e.g., HIPAA protocol)
6. Determine steps to make the repo public again (if appropriate)

2 Github Basics

This section covers the basics of how Github works. There are a lot of resources online that can also help you understand. Github is a website that is based on Git. Git is a common version control software.

2.1 What is Git

Git was developed to track changes in software over time, with multiple people working on the same code at once. You can download Git onto your PC and use it without worrying about Github at all. The diagram below shows how a project is developed using Git version control.



The project starts in an initial state, called the main (sometimes master) branch. In the diagram above, the main branch is colored green. Each circle is the code at a point in time, referred to as a commit. When a change is made to the code, you must choose to commit the change to the version history. You will also be asked to write a short description of what you changed or added. Uncommitted changes are considered staged, since they are waiting to be accepted or rejected.

For minor fixes, like correcting a typo, you could edit the main branch of code and commit the changes to it directly. But sometimes a change is more involved, and working on it will take time. Furthermore, sometimes you need to make changes to the foundations of a script that render it non-functional, but other people rely on the code to keep running. In this case, Git allows you to create a branch (in blue and orange above). Branches are a copy of the code that is separate from the main, functional code. You can then make edits to your branch without breaking the functional code. When you have finished work on the branch, you can then merge your changes into the main branch. In the diagram above, the blue branch took three commits to finish, whereas the orange commit only took one.

When making a commit, or merging a branch into main, you will have the opportunity to review the old version of the code and the new version. This is so each change is highlighted and looked over before it is added to the version history. The above process allows for complex projects with several contributors to be developed with a clear history of who did what and when, with the ability to go back to previous versions of the project, if need be.

A Git project can consist of one or more files that are located together in a folder. This location, and the files in it are known as a repository, or repo. All the files and subfolders in the repo are included in the project's version control. Git can be used by typing commands into a terminal or command prompt, but other tools (Github, Rstudio, etc.) provide a user interface that can make getting started easier.

Here are the key terms from the description above:

- Version control - Tracking, recording, and organizing changes to a project over time
- Git - a version control software
- Main - The base version of a project
- Branch - a copy of the main project, where changes can be made
- Staged Change - an edit to a file in a repo that has not been committed
- Commit - Officially adding a change to the version history (to the main project, or to a branch)
- Merge - Adding changes from a branch back into the main project
- Repository - A location that contains the files and folders that make up a project

2.2 What is Github

As stated above, you can use Git locally on your own computer. In this case, you are working on a local repository, or a project on your own computer. Github is a remote repository; a place to store the code that is not your own computer. This allows the code, with all its version