Github handbook

Hópur 24

Table of Contents

Introduction

- What is GitHub?

  GitHub is a web-based platform that uses Git for version control. It allows individuals and teams to collaborate on software development projects by providing tools for source code management, tracking changes, and facilitating collaboration. Users can create repositories to store and manage their code, track issues, and implement features such as branching and merging for efficient collaboration. GitHub has become a central hub for many open-source projects and serves as a widely used platform for sharing and developing software.
- What is GitHub Desktop, and how can it be used?

  GitHub Desktop is an open source application that helps you share work from files, GitHub or other sources. GitHub Desktop is used with other external tools e.g. VScode, to make changes to the code, by creating a new branch in GitHub.

Setting Up Accounts

Creating a GitHub account is a pretty straightforward process, done in a few steps.

1. **Visit the GitHub website:** Open your web browser and go to GitHub.
2. **Sign up:** On the GitHub homepage, you'll find a "Sign up" button. Click on it.
3. **Fill in your information:** You'll be prompted to enter some basic information.
    - E.g. Username and password.

4. **Choose a plan:** GitHub offers free plans for public repositories and paid plans for private repositories. Choose the plan that best suits your needs. If you're just getting started, the free plan is usually sufficient.
5. **Set up your profile:** You can customize your GitHub profile by adding a profile picture, bio, and other details. This step is optional but can help others identify you on GitHub.
6. **Explore GitHub:** Once your account is set up, you are ready to go. Follow other users, and start creating repositories to store and manage your code.

GitHub Desktop:

Download GitHub Desktop:

- Visit the GitHub Desktop download page.
- Click on the Download button (Windows or macOS).

Install GitHub Desktop:

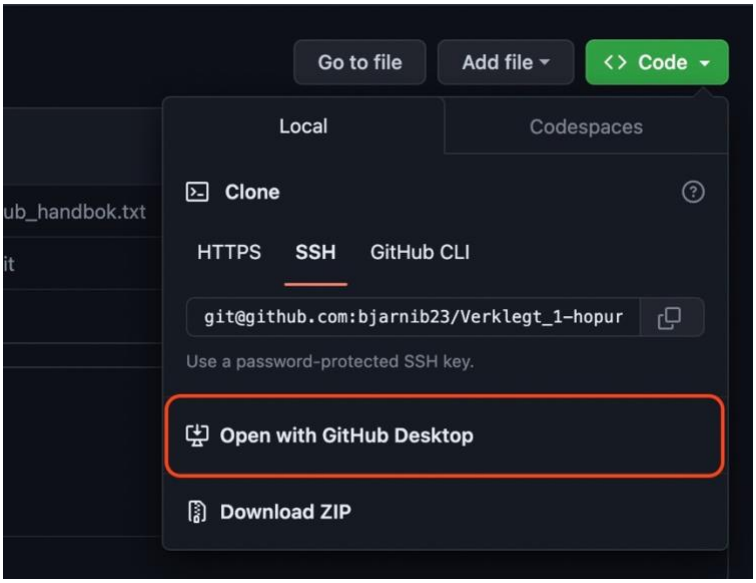
- Once the download is complete, run the installer and follow the on-screen instructions.
- Accept the license agreement and choose your preferences during the installation process.

Launch GitHub Desktop:

- After installation, launch GitHub Desktop.

Opening a repository on Github Desktop:

You can open your repository or a repository that has been shared to you in the Github Desktop app. You can do that by opening the repository you want to transfer to the app. There you can press the green “code” button, and there press the “Open with Github Desktop”.



Linking GitHub Desktop to Your GitHub Account:

1. **Sign in to your GitHub account:**
   - When you open GitHub Desktop, you'll be prompted to sign in.
2. **Enter your GitHub credentials:**
   - Provide your GitHub username and password and click "Sign in."

Configure Git settings:

- If you haven't configured your Git identity, GitHub Desktop may prompt you to enter your name and email address. This information is used to associate your commits with your GitHub account.

Finish setup:

- Once you've signed in and configured your Git settings, GitHub Desktop is ready to use.

Creating a Repository:

Create a new repository or clone an existing one:

- You can create a new repository on GitHub Desktop or clone an existing one from your GitHub account.

Start working:

- GitHub Desktop provides a user-friendly interface to manage changes, create branches, commit code, and sync with your GitHub account.

Core Operations

Cloning

- How to clone a project (repository) to your computer?
    - To make it easier to fix merge conflicts or push larger commits, you can clone a repository from GitHub to your computer or codingspace. To clone a repository, a repository is copied from GitHub to your computer, or to a remote virtual machine when a codespace is created.
      When a repository is cloned, GitHub retrieves a complete copy of all the repository data that is currently stored there, including all versions of all project files and folders. You have the option to pull changes from GitHub or submit your own modifications to the remote repository on GitHub

Commits

- How to make changes to the code and save them on GitHub?
    - To save changes that you've made to the code, you choose the branch where the changes are made, use the git functions or use Git Desktop to push the changes. When you make a commit, a commit message is included that describes the changes briefly.

Push

- The Git push command is used to transfer commits to your repository. Git push is one of four commands to update the local working branch. By default, the Git push command only updates the branch that you are checked out to.
    - When using Git push command, you can use the Git status command to check on what branch you are on
- How do I use Git push?

  You can use git pull in your terminal or in Github Desktop.

  Here below are the commands that you can use and also a short explanation on them:

  git push -f:

  Force a push that would otherwise be blocked, usually because it will delete or overwrite existing commits

  git push -u origin [branch]:

  Useful when pushing a new branch, this creates an upstream tracking branch with a lasting relationship to your local branch

  git push --all:

  Push all branches

  git push --tags:

  Publish tags that aren't yet in the remote repository

Pull

- The Git pull function is used to fetch new changes done on the local working branch. Git pull updates every remote tracking branch as well as your current local working branch.
- How to use Git pull?

  You can use git pull in your terminal or in Github Desktop.

  Here below are the commands that you can use and also a short explanation on them:

  Git pull --rebase:

  Updates your local working branch with remote commits, but prevents merge commits by rewriting history so local commits come after all new remote commits.

  Git pull --force:

  When using this command, you can compel a fetch of a certain remote tracking branch that might not otherwise be fetched because of conflicts. See the section below on using git reset to make Git replace your current branch in order to match the remote tracking branch.

  Git pull --all:

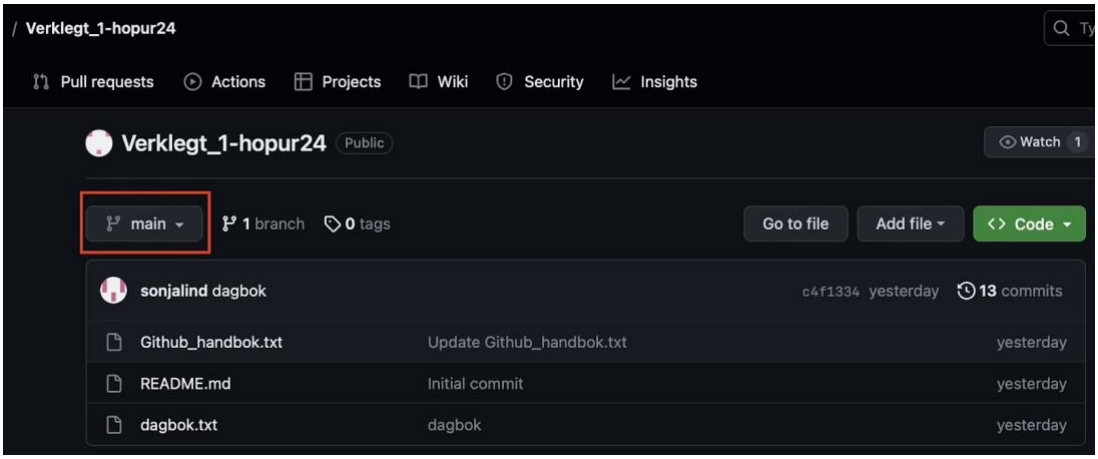
  This is a useful command when working on a fork or in another situation where there are several remotes.

Branching and Merging

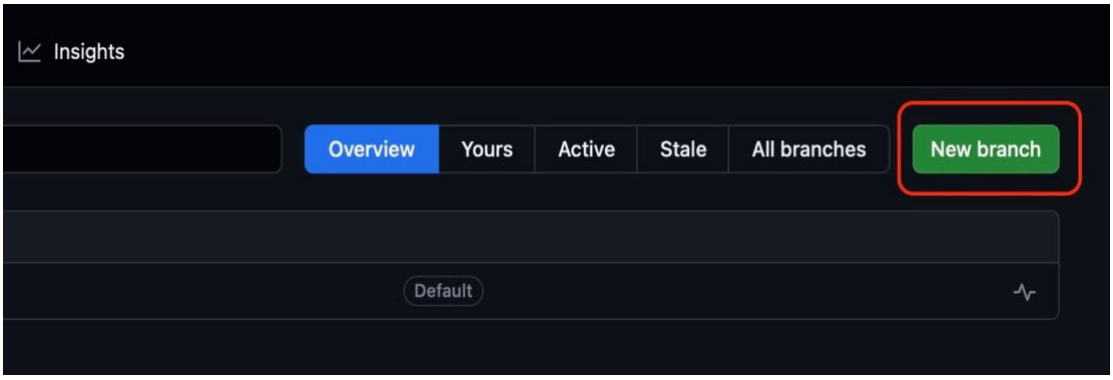
Creating a new branch

- On the main page in GitHub.com you click on a file tree tab on the left side of your

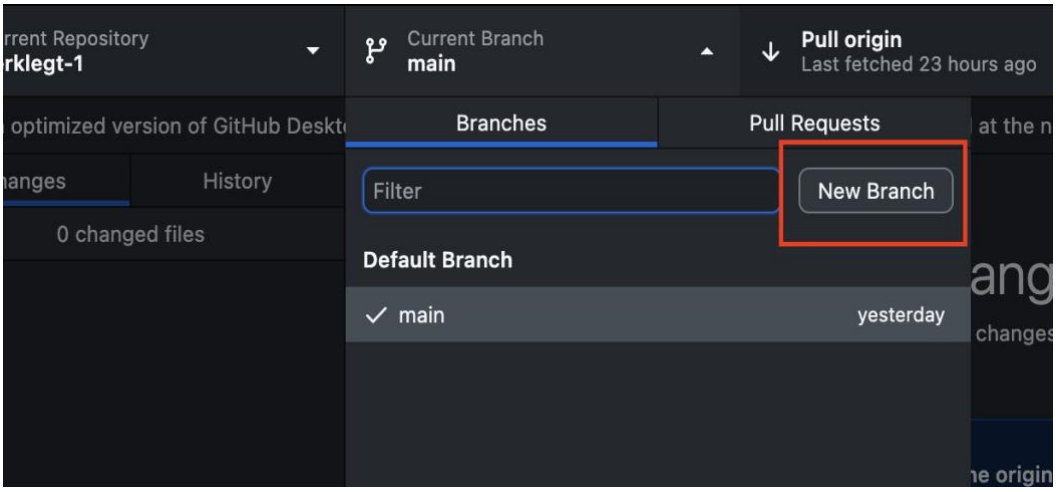
  screen.



From there you click on “view all branches”. Once you’ve opened all branches, click on “New branch”.

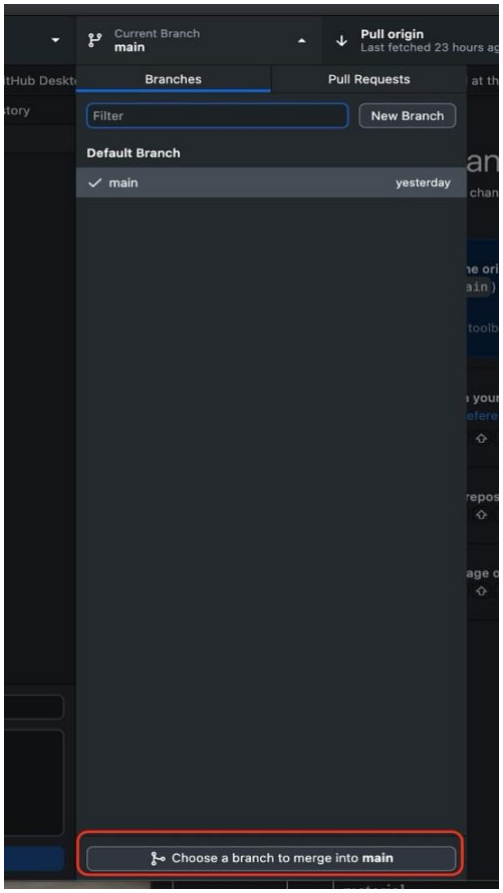


Another way is to use Github Desktop. On the main page, click on the file tree tap and then “New branch”



Merging branches with others

- To keep your local branch up to date with the remote repository, simply pull any new commits that have been made on the branch on GitHub since your last sync. This is especially important if you are working on the project from multiple devices or if multiple contributors are involved. Remember, pulling only updates your local version of the repository, so in order to update the branch on GitHub, you will need to push your changes.
  In GitHub Desktop, click on “current branch”(file tree tab), choose a branch that you’d want to merge and click on Choose a branch to merge into “BRANCH”



To push your local changes to the remote repository, in the repository bar, click **Push origin**.