Open Source Software Development

Ung Văn Giàu
Email: giau.ung@eiu.edu.vn

Lab 2
Distributed Version Control System

Contents

1. GitHub

Use **web interface** and **GitHub Desktop** to do the following:

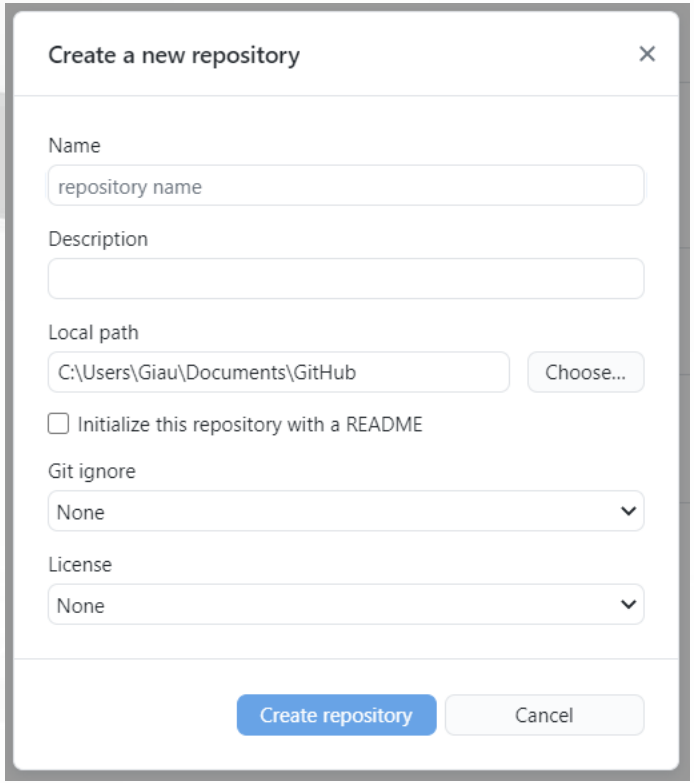
- Create and use a repository
- Start and manage a new branch
- Make changes to a file and push them to GitHub as commits
- Open and merge a pull request

GitHub Desktop

Basic

a. Creating a new repository

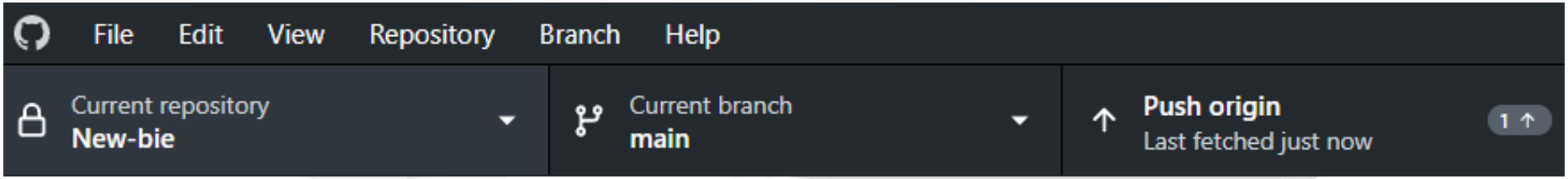
- Click File > New Repository…
- Fill in the fields and select options



- Click **Create repository**

b. Exploring GitHub Desktop

The GitHub Desktop menu bar

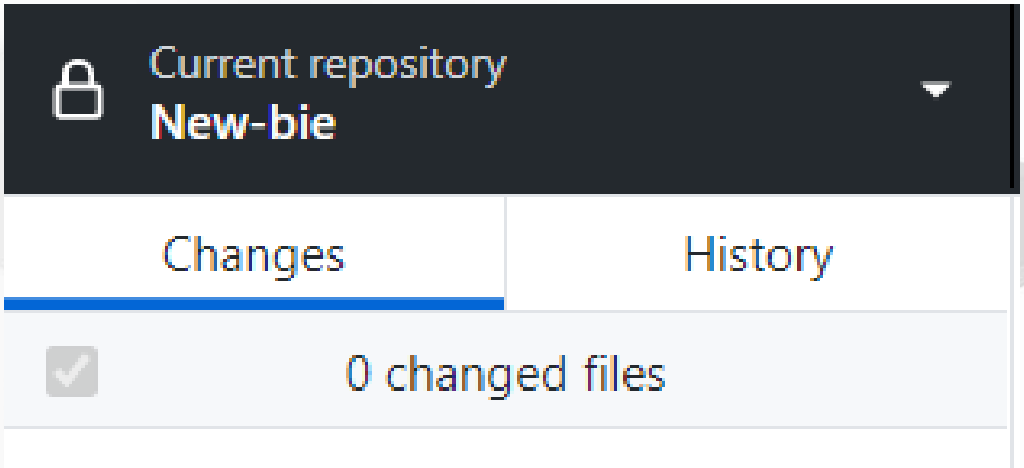


- **Current repository** shows the name of the repository you're working on
- **Current branch** shows the name of the branch you're working on.
- **Publish repository/Origin** appears because you haven't published your repository/changes to GitHub yet

This section of the bar will change based on the status of your current branch and repository.

b. Exploring GitHub Desktop

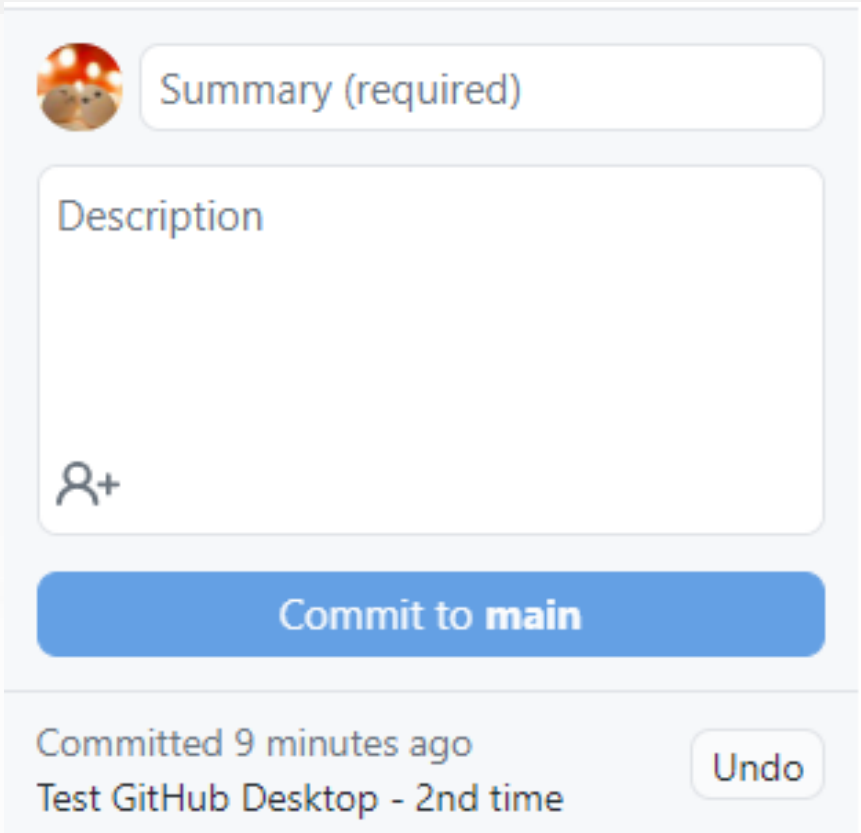
Changes and History



- The **Changes** view shows changes you've made to files in your current branch but haven't committed to your local repository
- The **History** view shows the previous commits on the current branch of your repository

b. Exploring GitHub Desktop

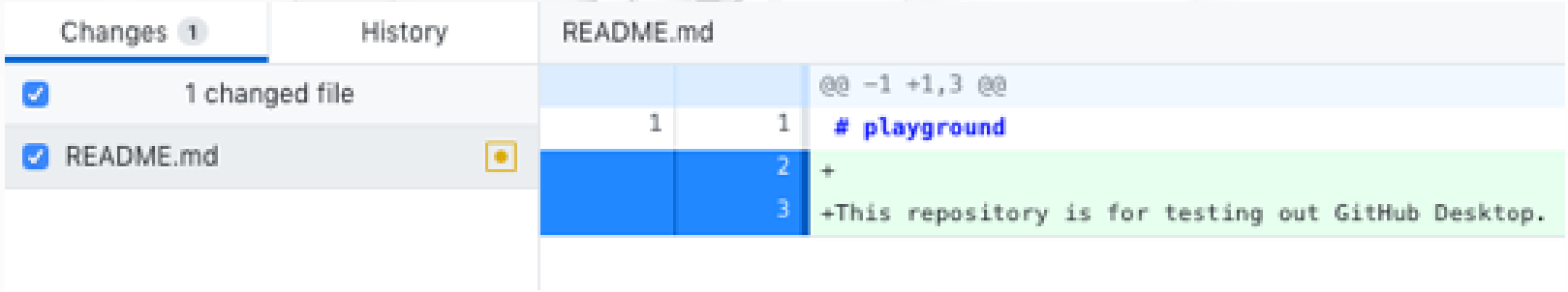
Changes and History



- This is where you'll commit new changes.

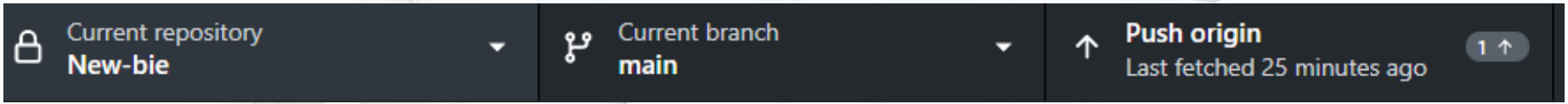
c. Making, committing, and pushing changes

- To launch an external editor from within GitHub Desktop, click **Repository**, then click **Open in EDITOR**
- Make some changes to the README.md file
- Navigate to the **Changes** view.
  - In the file list, you should see your *README.md*
  - The checkmark to the left indicates that the changes you've made to the file will be part of the commit



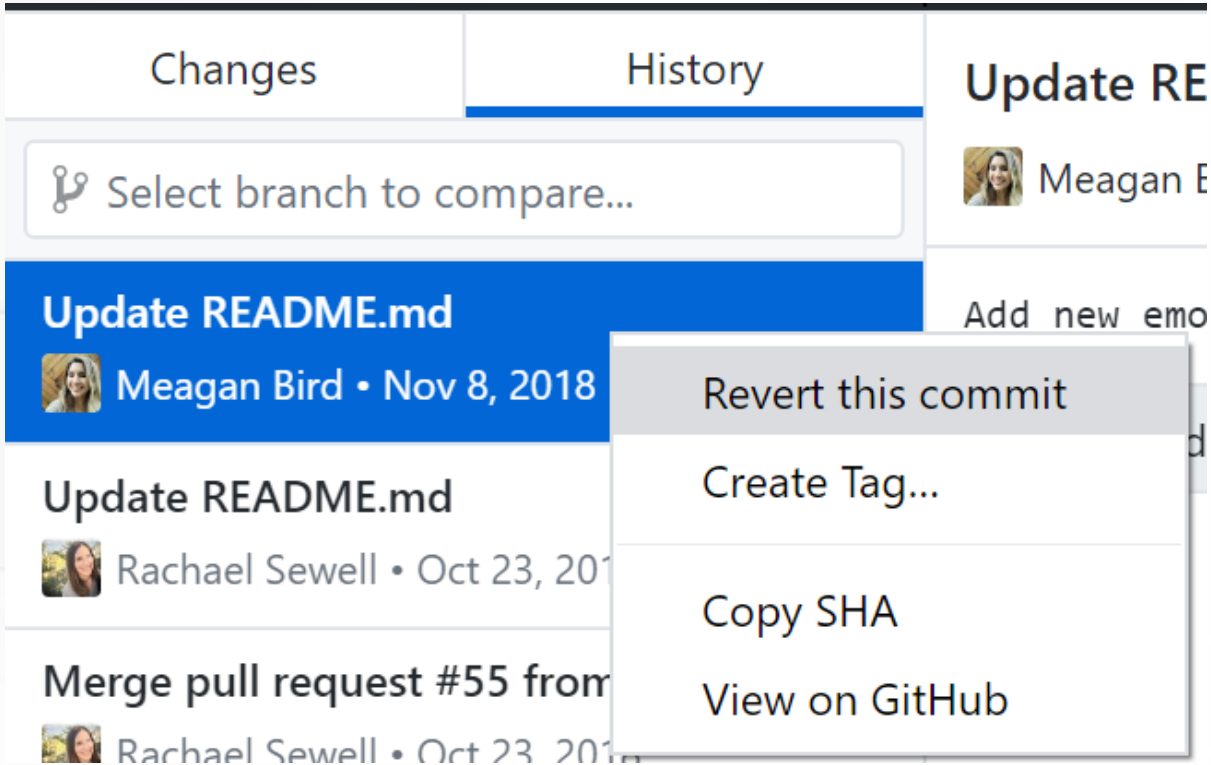
c. Making, committing, and pushing changes

- At the bottom of the **Changes** list, enter a commit message and Description
- Click **Commit to BRANCH NAME**
- To push changes to the remote repository on GitHub, click **Push origin**.



d. Reverting a commit

- Click **History**
- Right-click the commit you want to revert and click **Revert This Commit**.



2. Using Git Command Line to connect GitHub

Ubuntu 24.04

- Installing and Configuring Git
- Generating a new SSH key
- Adding a new SSH key to GitHub account
- Connecting to GitHub
- Creating a personal access token
- Using a token on the command line

Using Git Command Line to connect GitHub

Ubuntu 24.04

Installing and Configuring Git

- Get a GitHub account
- Install git

  sudo apt-get install git
- Set up git with username and email

  Open a terminal/shell and type:
  - $ git config --global user.name “Your name here”
  - $ git config --global user.email “your_email@example.com”

Generating a new SSH key

- Check if you have files *~/.ssh/id_rsa* and *~/.ssh/id_rsa.pub*

- If not, create such public/private keys:
  - Open a terminal/shell and type:

    $ ssh-keygen -t rsa -C "your_email@example.com"
  - When you’re prompted to "Enter a file in which to save the key," press Enter. This accepts the default file location
  - At the prompt, type a secure passphrase

Adding a new SSH key to GitHub account

- Copy the SSH public key (~/.ssh/id_rsa.pub) to your clipboard
- In the upper-right corner of GitHub page, click your profile photo, then click **Settings**
- In the user settings sidebar, click **SSH and GPG keys**
- Click **New SSH key** or **Add SSH key**
- In the “Title” field, add a descriptive label for the new key
- Paste your key into the “Key” field
- Click **Add SSH key**
- If prompted, confirm your GitHub password.

Connecting to GitHub

- In a terminal/shell, type the following to test it:

  $ ssh -T git@github.com

- If it says something like the following, it worked:

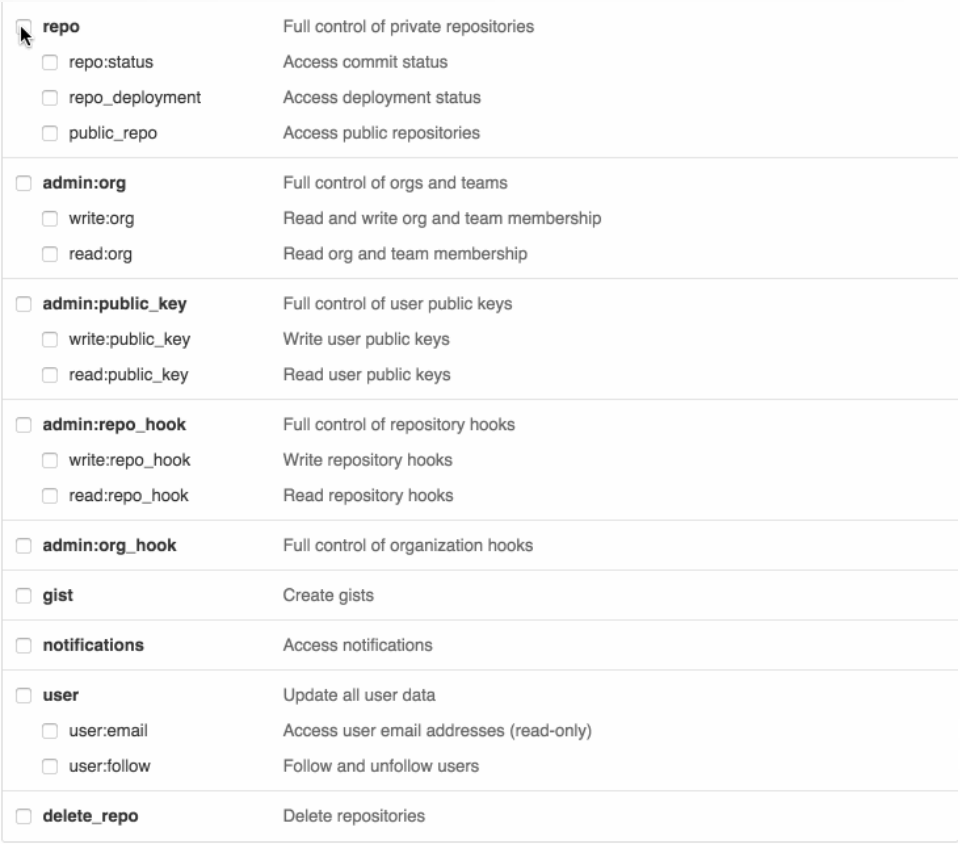
  Hi username! You've successfully authenticated, but Github does not provide shell access.

Creating a personal access token

- In the upper-right corner of GitHub page, click your profile photo, then click **Settings**
- In the left sidebar, click **Developer settings**
- In the left sidebar, click **Personal access tokens**
- Click **Generate new token**
- Give your token a descriptive name
- To give your token an expiration, select the **Expiration** drop-down menu, then click a default or use the calendar picker

Creating a personal access token

- Select the scopes, or permissions, you'd like to grant this token. To use your token to access repositories from the command line, select **repo**. Then, Click **Generate token**



Using a token on the command line

- Once you have a token, you can enter it instead of password when performing Git operations over HTTPS
- For example:
  - $ git clone https://github.com/username/repo.git
  - Username: your_username
  - Password: your_token

3. Basic Git commands

- Clone your repo Modify files
- Add, Commit and push them to repo

Basic Git commands

Clone repository to local machine

- From repository page, click the green button labeled **Clone or download**, and in the “Clone with HTTPs” section, copy the URL for your repository
- On local machine, open **terminal** and change current working directory to the location where you would like to clone your repository
- Type:

  git clone https://github.com/URL-TO-REPO-HERE

Tracking changes with git add and git commit

- Move to your repo: cd your_repo
- To see all files: ls -a

- Edit a file in your repo:
  - Use text editor to open and make a few edits to any file
  - Type: git status to see the files you have modified
  - To keep track of this change to this file, you need to
    - ✓ add the changes, then
    - ✓ commit the changes.

Add and commit changes

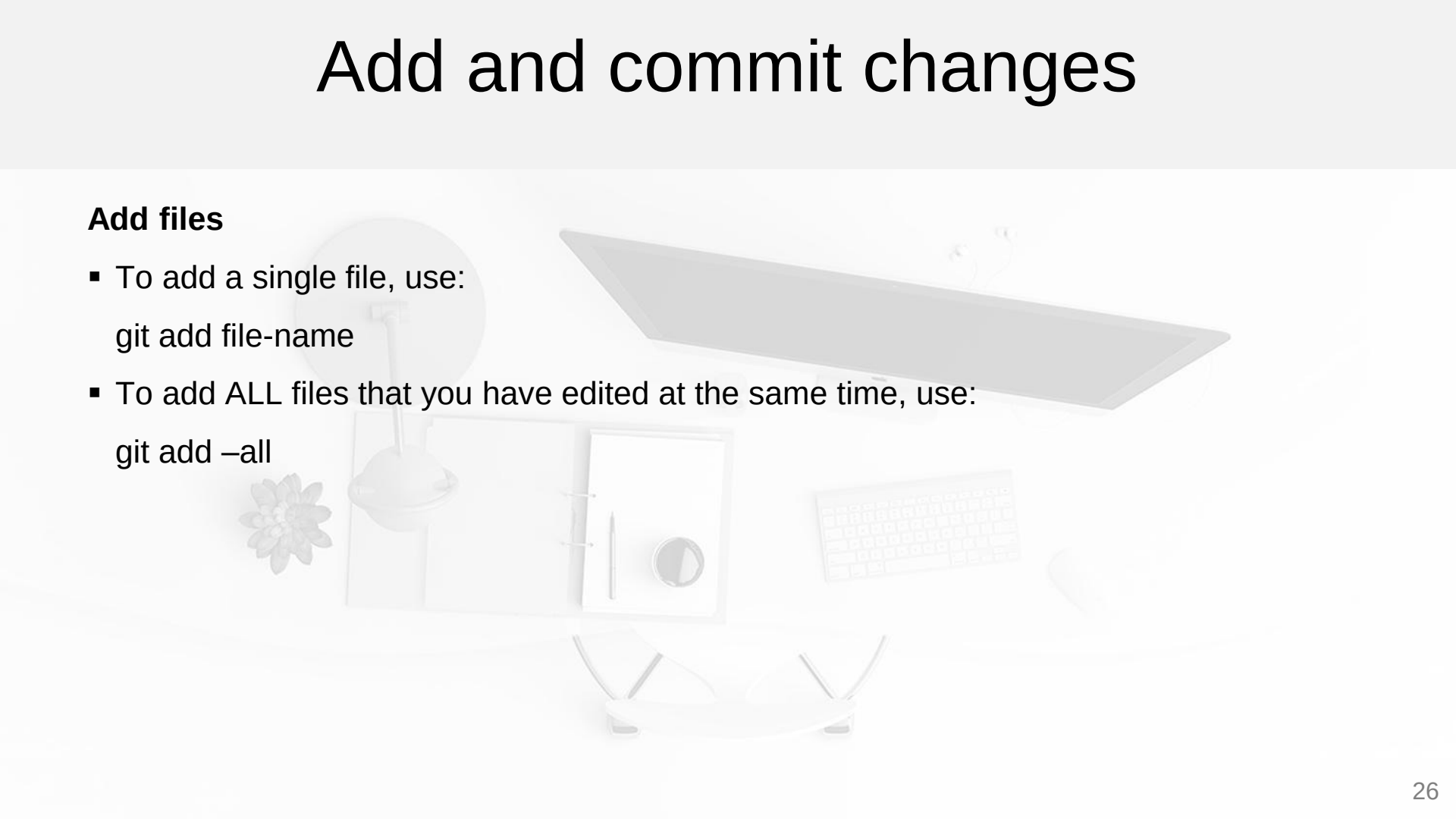
Add files

- To add a single file, use:

  git add file-name

- To add ALL files that you have edited at the same time, use:

  git add –all



Add and commit changes

Commit files

If you are not committing a lot of changes, you can create a short one line commit message using the -m flag:

```
git commit -m “Your message”
```

Push changes to GitHub

- To push your changes to GitHub, type:

  git push

- You will then be prompted for your GitHub username and password (use token instead of token).

4. Creating README

https://github.com/PHPMailer/PHPMailer#installation--loading

Contributing

Please submit bug reports, suggestions and pull requests to the GitHub issue tracker.

We're particularly interested in fixing edge-cases, expanding test coverage and updating translations.

If you found a mistake in the docs, or want to add something, go ahead and amend the wiki - anyone can edit it.

If you have git clones from prior to the move to the PHPMailer GitHub organisation, you'll need to update any remote URLs referencing the old GitHub location with a command like this from within your clone:

```
git remote set-url upstream https://github.com/PHPMailer/PHPMailer.git
```

Please *don't* use the SourceForge or Google Code projects any more; they are obsolete and no longer maintained.

Sponsorship

Development time and resources for PHPMailer are provided by Smartmessages.net, a powerful email marketing system.

SMARTMESSAGES

Other contributions are gladly received, whether in beer, T-shirts, Amazon wishlist raids, or cold, hard cash. If you'd like to donate to say "thank you" to maintainers or contributors, please contact them through individual profile pages via the contributors page.

Changelog

See changelog.

History

- PHPMailer was originally written in 2001 by Brent R. Matzelle as a SourceForge project.
- Marcus Bointon (coolbru on SF) and Andy Prevost (codeworxtech) took over the project in 2004.
- Became an Apache incubator project on Google Code in 2010, managed by Jim Jagielski.
- Marcus created his fork on GitHub in 2008.

5. Creating a Wiki

GitHub web interface

GitHub supports many extras in Markdown that help you reference and link to people. If you ever want to direct a comment at someone, you can prefix their name with an @ symbol: Hey @kneath — love your sweater!

But I have to admit, tasks lists are my favorite:

- [x] This is a complete item
- [ ] This is an incomplete item

When you include a task list in the first comment of an Issue, you will see a helpful progress bar in your list of issues. It works in Pull Requests, too!

And, of course emoji!

6. Creating Boards

GitHub web interface