

暑期实训学习报告

1813012 柳鑫

一. 学习内容

上午:

- ① Git 安装
- ② 创建版本库
- ③ 版本回退
- ④ 工作区与暂存区
- ⑤ 管理修改
- ⑥ 撤销修改
- ⑦ 删除文件

下午:

- ① 使用 GitHub
- ② 远程仓库
- ③ 分支管理
- ④ 标签管理
- ⑤ 忽略特殊文件
- ⑥ 配置别名
- ⑦ 使用 SourceTree

二. 学习结果

上午:

① Git 安装

在官方网站下载安装包之后，安装 Git。

```
MINGW64~/Users/86332/Desktop
$ git
usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]
      [--exec-path<path>] [--html-path] [--man-path] [--info-path]
      [-p | --paginate] [-P | --no-pager] [--no-replace-objects] [--bare]
      [--git-dir<path>] [--work-tree<path>] [--namespace<name>]
      <command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
  clone      Clone a repository into a new directory
  init       Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
  add        Add file contents to the index
  mv         Move or rename a file, a directory, or a symlink
  restore    Restore working tree files
  rm         Remove files from the working tree and from the index
  sparse-checkout  Initialize and modify the sparse-checkout

examine the history and state (see also: git help revisions)
  bisect     Use binary search to find the commit that introduced a bug
  diff       Show changes between commits, commit and working tree, etc
  grep       Print lines matching a pattern
  log        Show commit logs
  show       Show various types of objects
  status     Show the working tree status

grow, mark and tweak your common history
  branch     List, create, or delete branches
  commit     Record changes to the repository
  merge      Join two or more development histories together
  rebase     Reapply commits on top of another base tip
  reset      Reset current HEAD to the specified state
  switch     Switch branches
  tag        Create, list, delete or verify a tag object signed with GPG

collaborate (see also: git help workflows)
  fetch      Download objects and refs from another repository
  pull       Fetch from and integrate with another repository or a local branch
  push       Update remote refs along with associated objects

'git help -a' and 'git help -p' list available subcommands and some
concept guides. See 'git help <command>' or 'git help <concept>'
to read about a specific subcommand or concept.
see 'git help git' for an overview of the system.
MINGW64~/Users/86332/Desktop
$
```

② 创建版本库

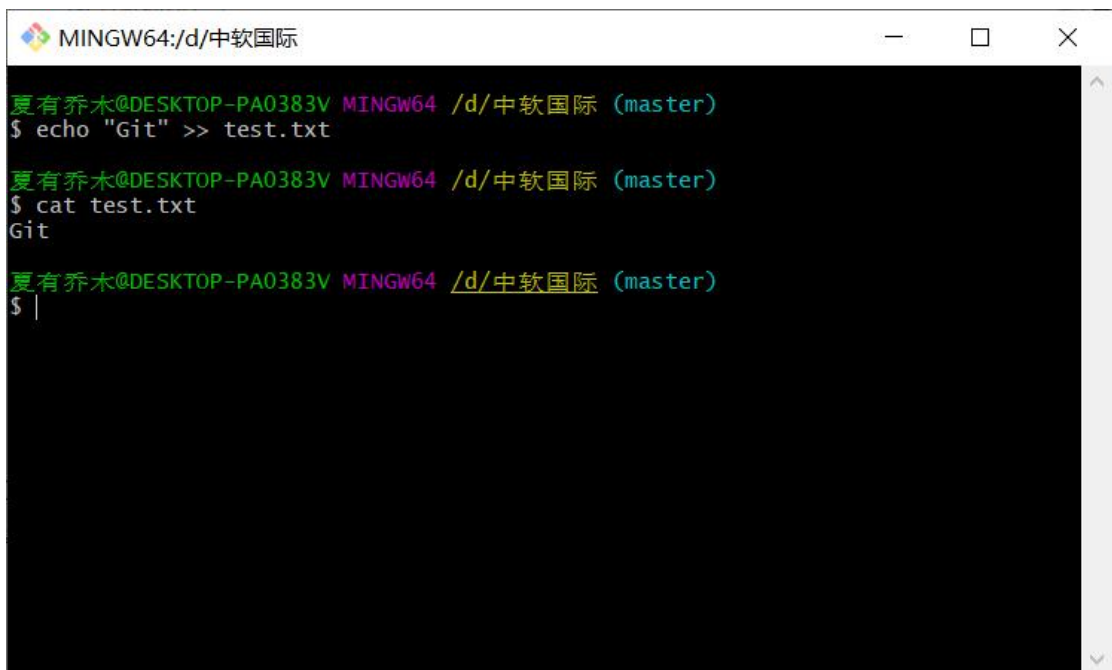
输入如下的代码，创建版本库。

```
MINGW64:/d/中软国际
$ git init
Initialized empty Git repository in D:/中软国际/.git/

MINGW64:/d/中软国际 (master)
$
```

③ 工作区与暂存区

输入如下的代码，在工作区创建 `test.txt` 文件，并输入信息。

A terminal window titled 'MINGW64:/d/中软国际' with standard window controls. The prompt is '夏有乔木@DESKTOP-PA0383V MINGW64 /d/中软国际 (master)'. The user enters '\$ echo "Git" >> test.txt'. The prompt repeats, and the user enters '\$ cat test.txt'. The output 'Git' is displayed. The prompt repeats again, and the user enters '\$ |' followed by a cursor.

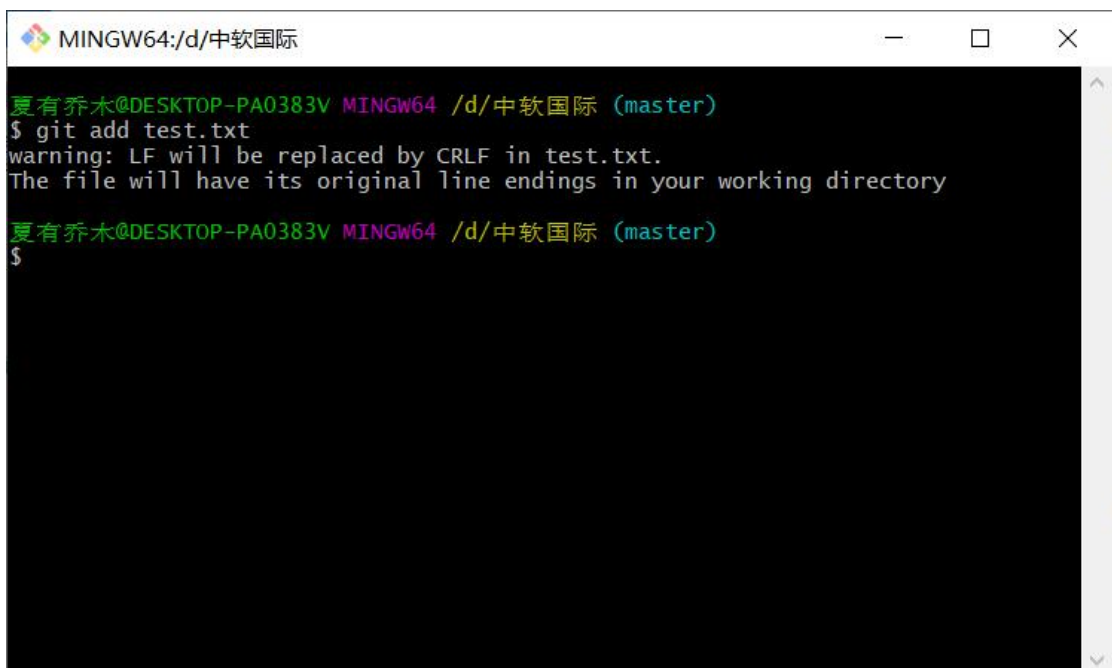
```
MINGW64:/d/中软国际

夏有乔木@DESKTOP-PA0383V MINGW64 /d/中软国际 (master)
$ echo "Git" >> test.txt

夏有乔木@DESKTOP-PA0383V MINGW64 /d/中软国际 (master)
$ cat test.txt
Git

夏有乔木@DESKTOP-PA0383V MINGW64 /d/中软国际 (master)
$ |
```

输入如下的代码，将 `test.txt` 文件提交到暂存区。

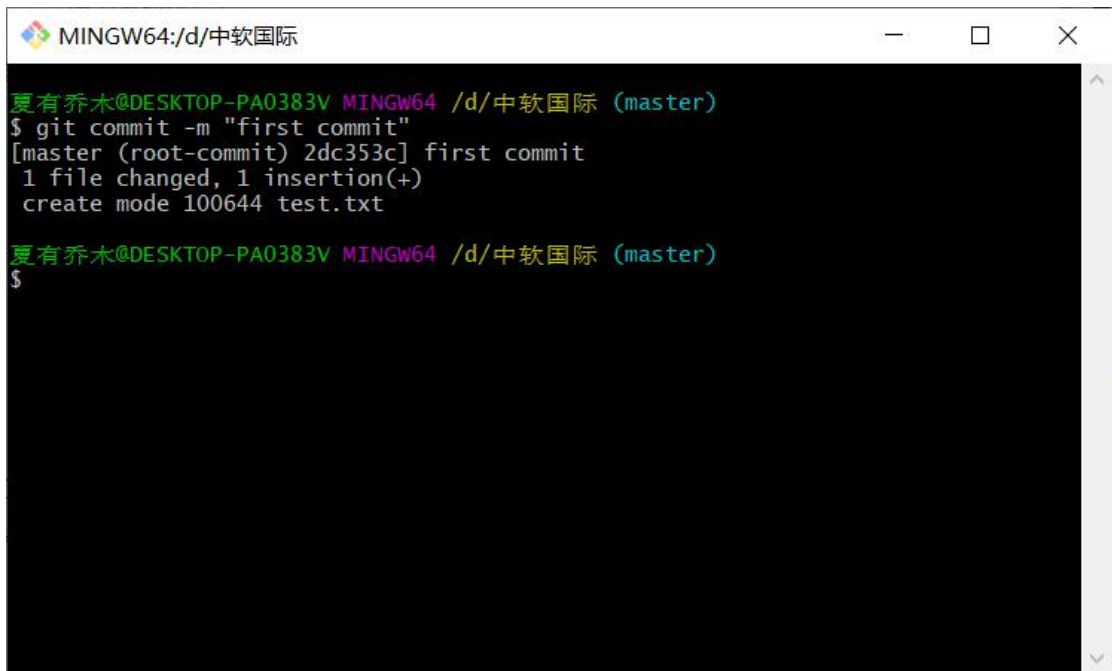
A terminal window titled 'MINGW64:/d/中软国际' with standard window controls. The prompt is '夏有乔木@DESKTOP-PA0383V MINGW64 /d/中软国际 (master)'. The user enters '\$ git add test.txt'. The output 'warning: LF will be replaced by CRLF in test.txt. The file will have its original line endings in your working directory' is displayed. The prompt repeats, and the user enters '\$' followed by a cursor.

```
MINGW64:/d/中软国际

夏有乔木@DESKTOP-PA0383V MINGW64 /d/中软国际 (master)
$ git add test.txt
warning: LF will be replaced by CRLF in test.txt.
The file will have its original line endings in your working directory

夏有乔木@DESKTOP-PA0383V MINGW64 /d/中软国际 (master)
$
```

输入如下的代码，将 `test.txt` 文件提交到版本库。



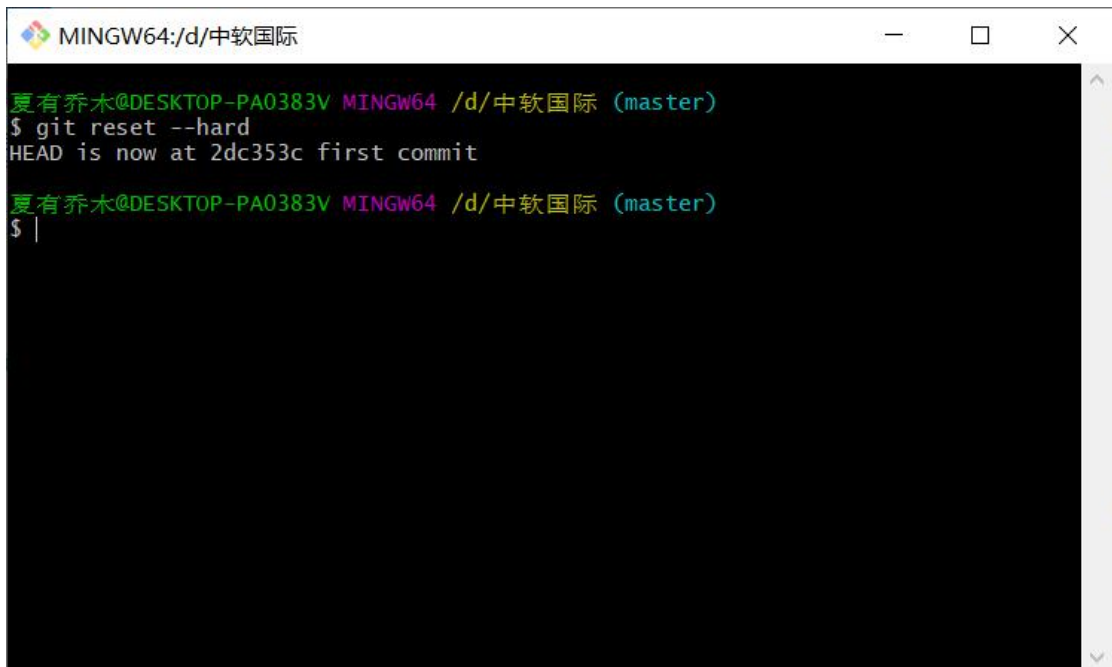
```
MINGW64:/d/中软国际

夏有乔木@DESKTOP-PA0383V MINGW64 /d/中软国际 (master)
$ git commit -m "first commit"
[master (root-commit) 2dc353c] first commit
1 file changed, 1 insertion(+)
create mode 100644 test.txt

夏有乔木@DESKTOP-PA0383V MINGW64 /d/中软国际 (master)
$
```

④ 版本回退

输入如下的代码，将版本库回退到第一次提交时的状态。



```
MINGW64:/d/中软国际

夏有乔木@DESKTOP-PA0383V MINGW64 /d/中软国际 (master)
$ git reset --hard
HEAD is now at 2dc353c first commit

夏有乔木@DESKTOP-PA0383V MINGW64 /d/中软国际 (master)
$ |
```

⑤ 管理修改

输入如下的代码，对版本库进行修改。

```
MINGW64:/d/中软国际

夏有乔木@DESKTOP-PA0383V MINGW64 /d/中软国际 (master)
$ git add test.txt
warning: LF will be replaced by CRLF in test.txt.
The file will have its original line endings in your working directory

夏有乔木@DESKTOP-PA0383V MINGW64 /d/中软国际 (master)
$ git commit -m "second commit"
[master cc6ddea] second commit
1 file changed, 1 insertion(+), 1 deletion(-)

夏有乔木@DESKTOP-PA0383V MINGW64 /d/中软国际 (master)
$
```

输入如下的代码，管理版本库的修改。

```
MINGW64:/d/中软国际

夏有乔木@DESKTOP-PA0383V MINGW64 /d/中软国际 (master)
$ git log
commit cc6ddea4161e5610d604ac7fcabdf15516b81d41 (HEAD -> master)
Author: 863320295 <863320295@qq.com>
Date: Mon Jun 15 16:12:13 2020 +0800

    second commit

commit 2dc353c860e285f1e53d8a86e784469188f675a9
Author: 863320295 <863320295@qq.com>
Date: Mon Jun 15 16:01:21 2020 +0800

    first commit

夏有乔木@DESKTOP-PA0383V MINGW64 /d/中软国际 (master)
$
```

⑥ 撤销修改

输入如下的代码，查看版本库的修改日志。

```
MINGW64:/d/中软国际
夏有乔木@DESKTOP-PA0383V MINGW64 /d/中软国际 (master)
$ git log
commit cc6dde4161e5610d604ac7fcabdf15516b81d41 (HEAD -> master)
Author: 863320295 <863320295@qq.com>
Date: Mon Jun 15 16:12:13 2020 +0800

    second commit

commit 2dc353c860e285f1e53d8a86e784469188f675a9
Author: 863320295 <863320295@qq.com>
Date: Mon Jun 15 16:01:21 2020 +0800

    first commit

夏有乔木@DESKTOP-PA0383V MINGW64 /d/中软国际 (master)
$
```

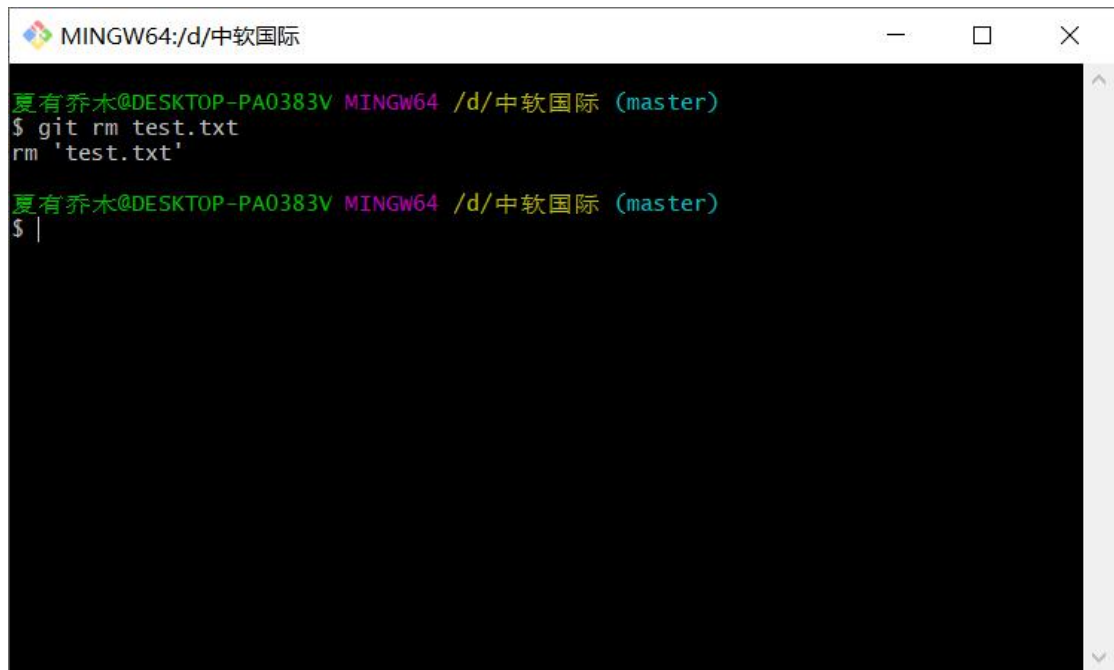
输入如下的代码，撤销对版本库的修改。

```
MINGW64:/d/中软国际
夏有乔木@DESKTOP-PA0383V MINGW64 /d/中软国际 (master)
$ git reset --hard 2dc353c860e285f1e53d8a86e784469188f675a9
HEAD is now at 2dc353c first commit

夏有乔木@DESKTOP-PA0383V MINGW64 /d/中软国际 (master)
$
```

⑦ 删除文件

输入如下的代码，删除 test.txt 文件。



```
MINGW64:/d/中软国际

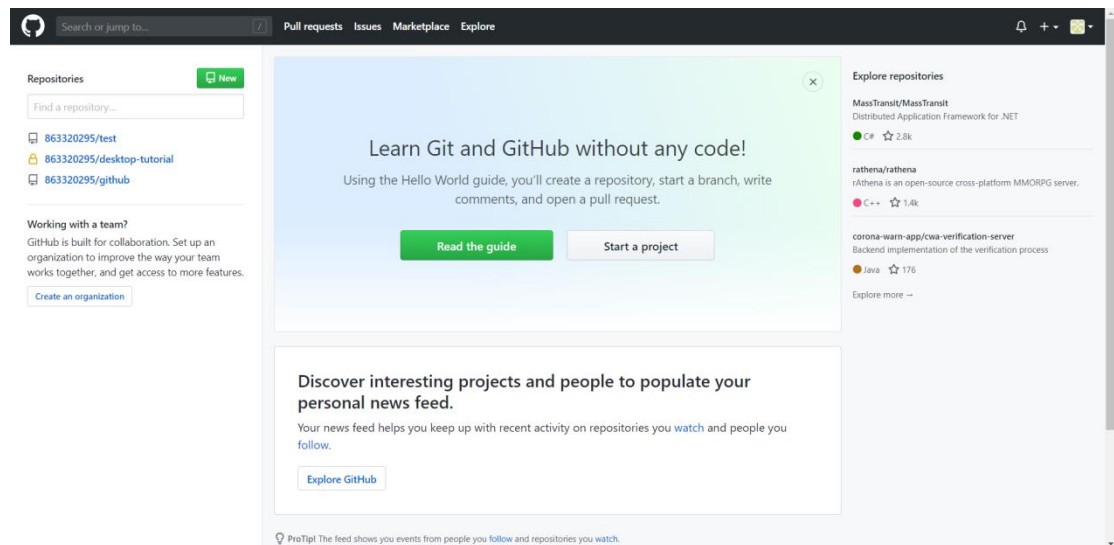
夏有乔木@DESKTOP-PA0383V MINGW64 /d/中软国际 (master)
$ git rm test.txt
rm 'test.txt'

夏有乔木@DESKTOP-PA0383V MINGW64 /d/中软国际 (master)
$ |
```

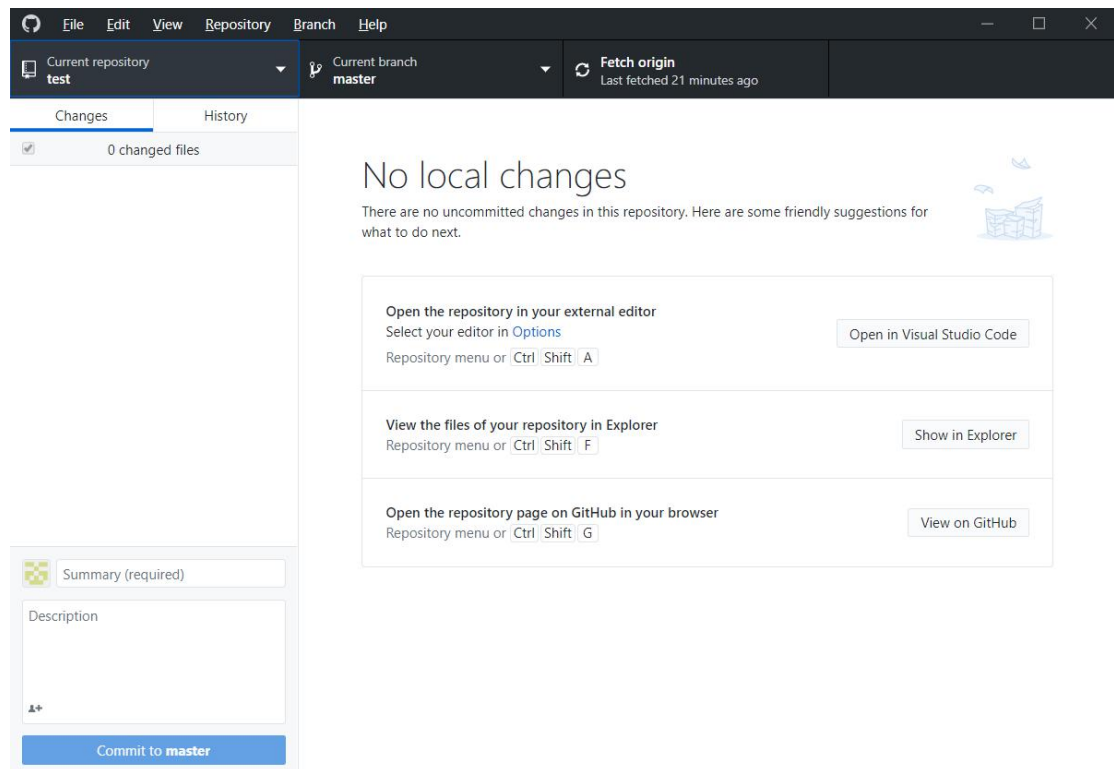
下午:

① 使用 GitHub

修改 hosts 文件之后，登录 GitHub 官方网站，注册账号。

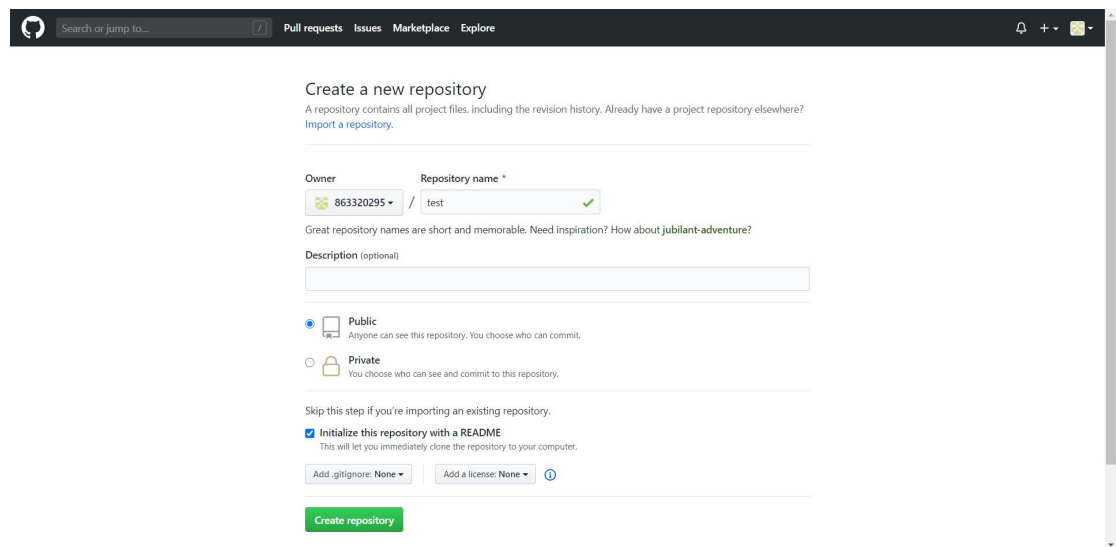


下载 GitHub 客户端，完成安装。



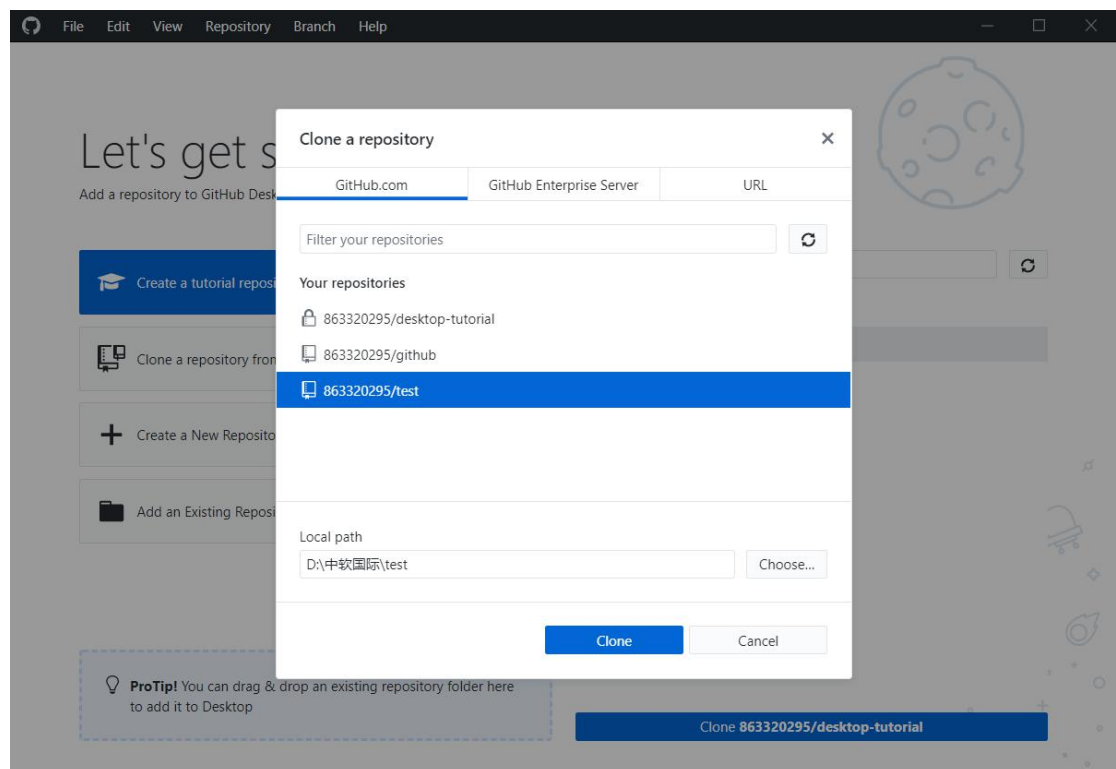
② 远程仓库

按照以下操作，创建远程仓库。



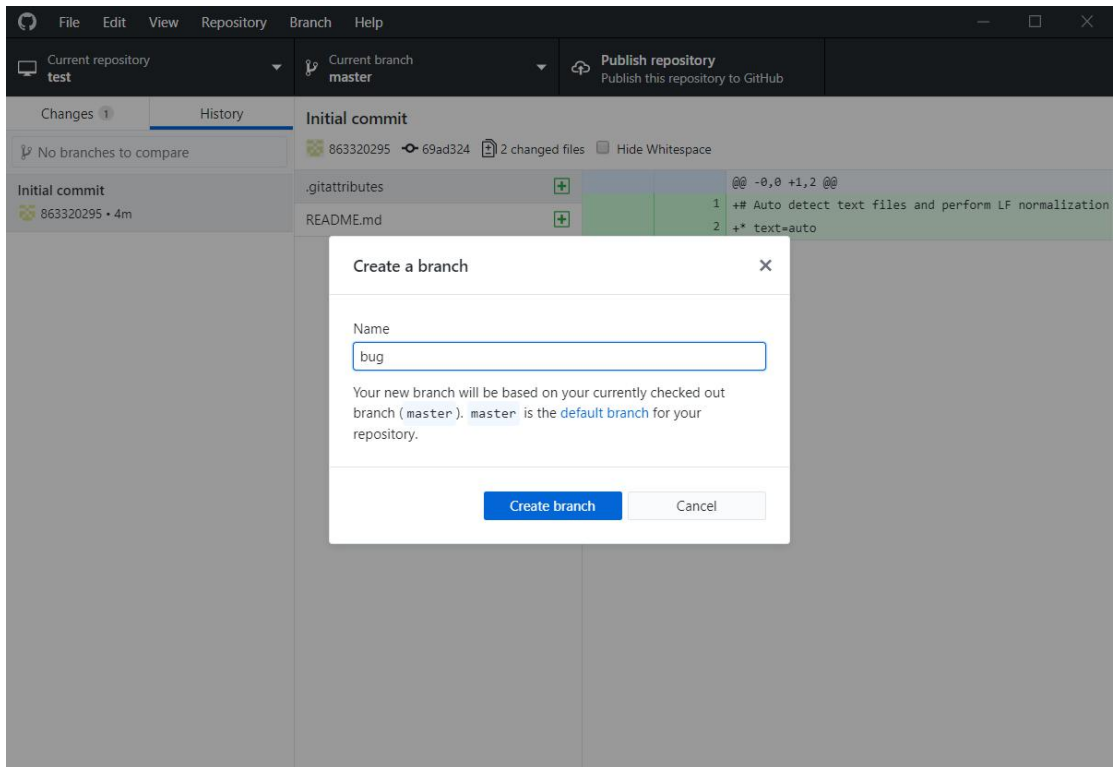
The screenshot shows the GitHub 'Create a new repository' page. At the top, there's a navigation bar with 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. Below the navigation bar, the page title is 'Create a new repository'. A sub-header says 'A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)'. The form has two main sections: 'Owner' and 'Repository name'. The 'Owner' is set to '863320295' and the 'Repository name' is 'test'. Below this, there's a note: 'Great repository names are short and memorable. Need inspiration? How about jubilant-adventure?'. There's a 'Description (optional)' text area. The 'Visibility' section has two options: 'Public' (selected) and 'Private'. Below this, there's a note: 'Skip this step if you're importing an existing repository.' The 'Initialize this repository with a README' checkbox is checked. At the bottom, there are two dropdown menus: 'Add .gitignore: None' and 'Add a license: None'. A green 'Create repository' button is at the bottom.

将远程仓库克隆到本地。

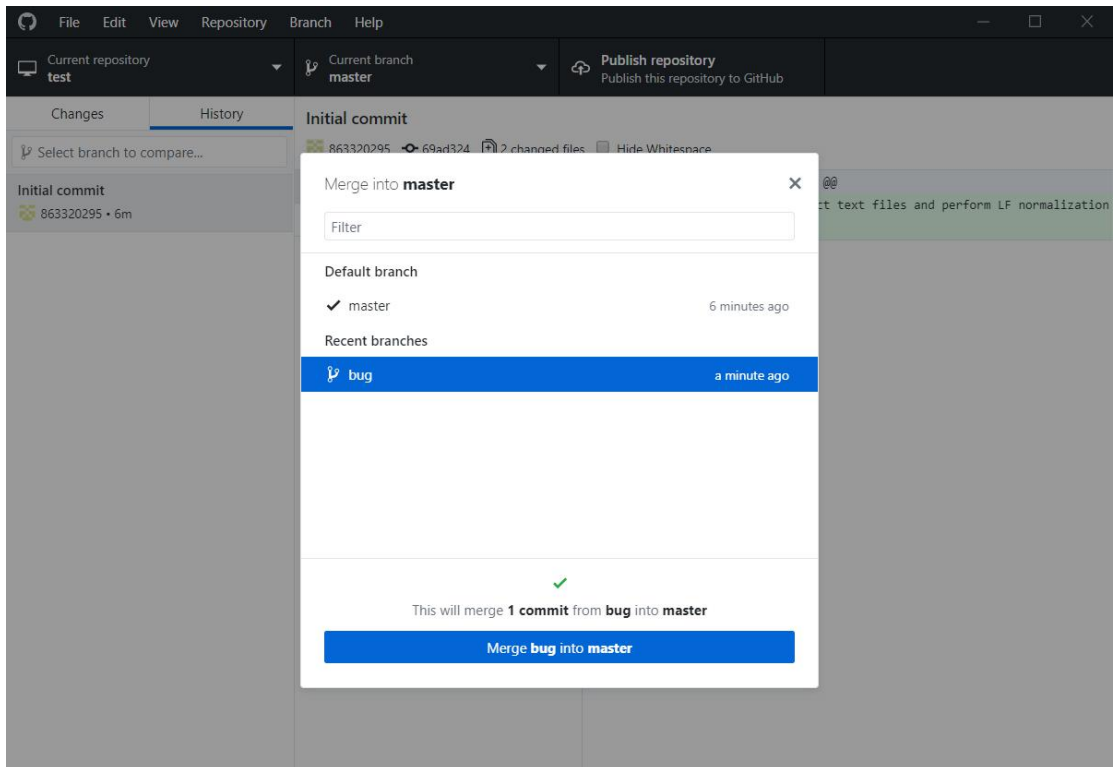


③ 分支管理

按照以下操作，创建新的分支。



在完成修改之后，将分支和 **master** 合并。



④ 标签管理

输入以下的代码，添加新的标签。

```
MINGW64:/d/中软国际/test

夏有乔木@DESKTOP-PA0383V MINGW64 /d/中软国际/test (master)
$ git tag v1.0

夏有乔木@DESKTOP-PA0383V MINGW64 /d/中软国际/test (master)
$ git tag
v1.0

夏有乔木@DESKTOP-PA0383V MINGW64 /d/中软国际/test (master)
$ git push origin v1.0
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/863320295/test.git
 * [new tag]          v1.0 -> v1.0

夏有乔木@DESKTOP-PA0383V MINGW64 /d/中软国际/test (master)
$ |
```

⑤ 忽略特殊文件

输入以下的代码，查询被忽略的特殊文件。

```
MINGW64:/d/中软国际/test

夏有乔木@DESKTOP-PA0383V MINGW64 /d/中软国际/test (master)
$ touch .gitignore|
```

⑥ 配置别名

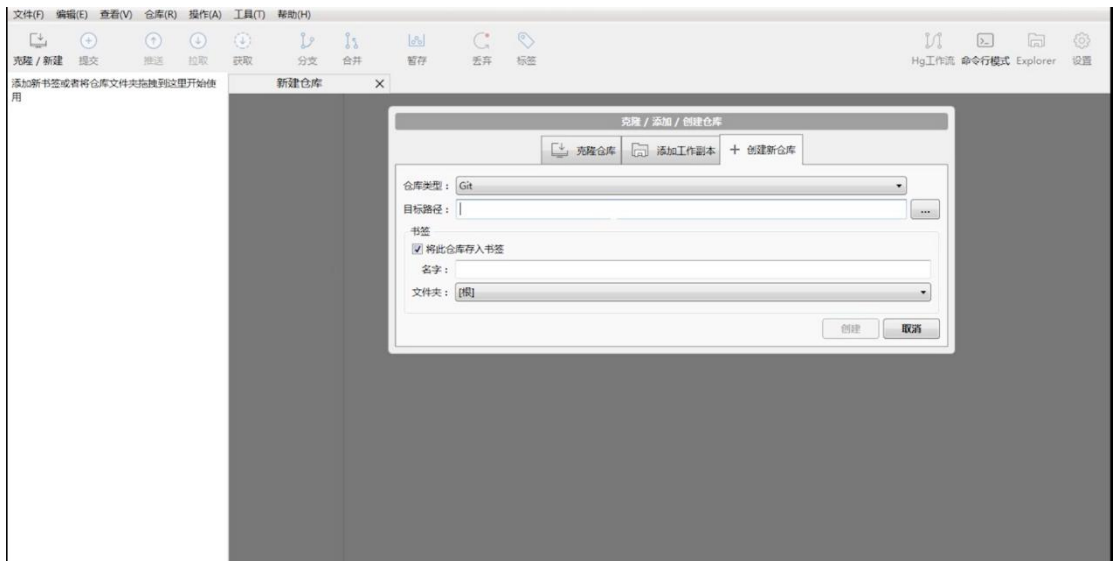
输入以下的代码，配置 git 指令的别名。

```
MINGW64:/d/中软国际/test

夏有乔木@DESKTOP-PA0383V MINGW64 /d/中软国际/test (master)
$ git config --global alias
```

⑦ 使用 SourceTree

在官方网站下载安装包之后，安装 SourceTree，进行使用。



三. 心得体会

经过一天的学习，我了解到 Git 作为一款开源的分布式版本控制系统，能够敏捷高效地处理任何大小的项目。利用 Git，我们可以在项目开发的过程中，方便地修改代码、提交代码，能够与别人进行有效率的合作。

而且，在分别使用 GitHub 客户端与 SourceTree 之后，对于我来说，GitHub 客户端操作起来更加舒适，我更加偏向于使用 GitHub 客户端来进行操作。