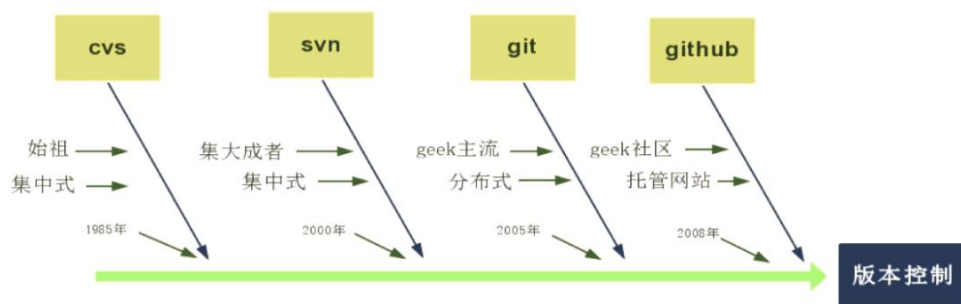


1、简介

备份文件
记录历史
回到过去
多端共享
团队合作



2、Linux（CentOS）安装

```
# yum install git
```

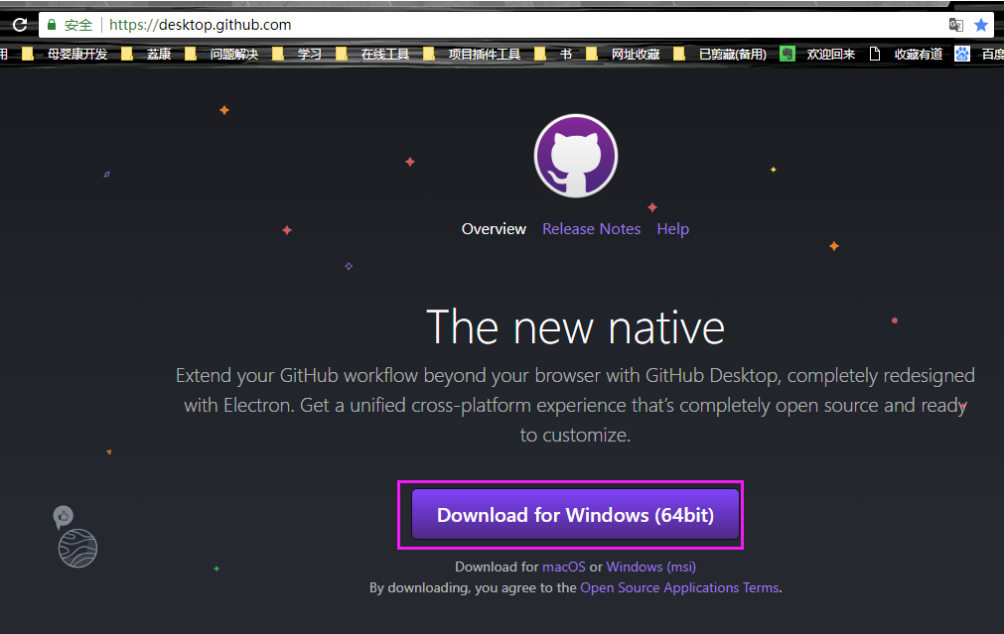
确认一下，y

```
git --help // 安装完成后，帮助命令
```

```
[root@localhost ~]# git --version
git version 1.8.3.1
```

3、Windows 下载

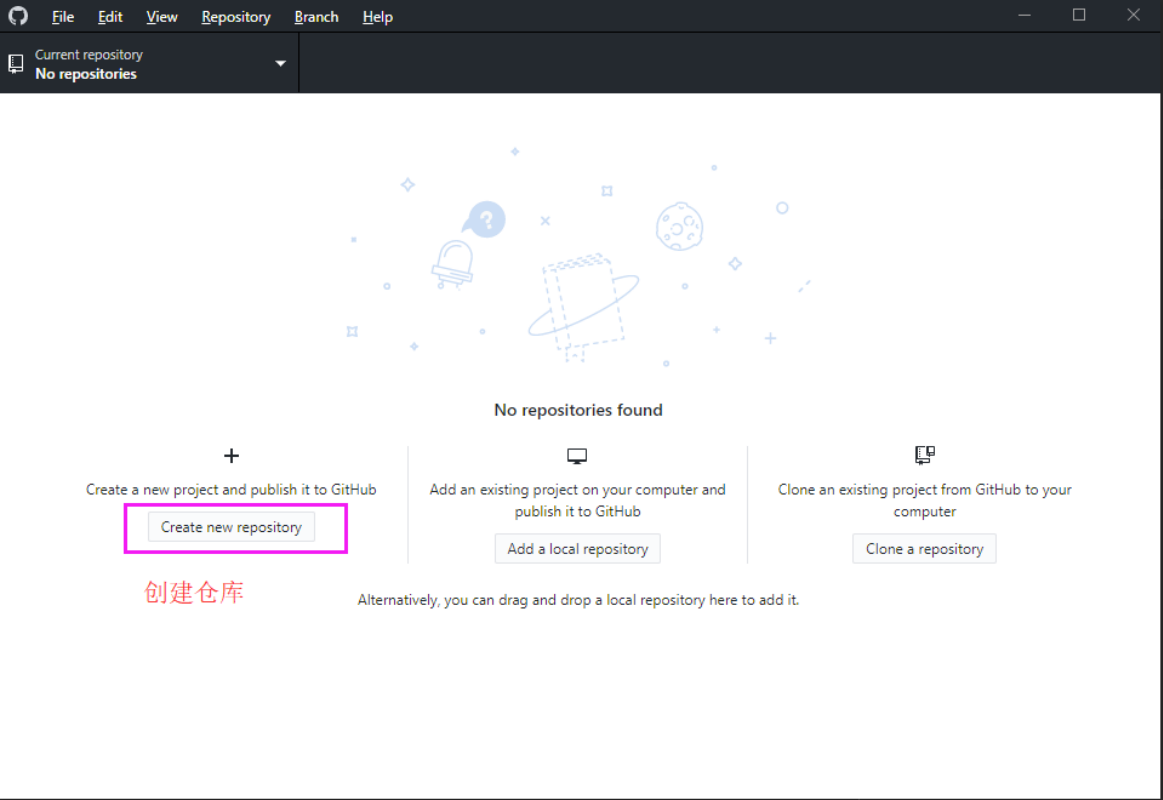
Windows 下载、安装



直接打开，自动会安装到 C 盘，什么都不用操作，跟谷歌浏览器一样无语，不过我去 C 盘找到它，剪切到了 D 盘，还好可以用。

此电脑 > Software (D:) > Software > GitHubDesktop				
名称	修改日期	类型	大小	
app-1.3.4	18/8/31 16:54	文件夹		
bin	18/8/31 16:55	文件夹		
packages	18/8/31 16:55	文件夹		
app.ico	18/8/31 16:52	看图王 ICO 图片...	126 KB	
GitHubDesktop.exe	18/8/31 16:51	应用程序	374 KB	
SquirrelSetup.log	18/8/31 16:55	LOG 文件	3 KB	
Update.exe	18/8/23 16:08	应用程序	1,494 KB	

打开之后我什么都不填写，进入到了这个界面，这些东西变化太快，看着办好了



Create a new repository

Name

HelloWorld

Description

第一个版本管理

Local path

E:\SoftwareData\GitHubRepository

仓库目录

Choose...

☒ Initialize this repository with a README

是否创建
Readme文件

Git ignore

None

忽略什么

License

None

开源协议

Create repository

Cancel

进入该界面

File Edit View Repository Branch Help

Current repository
HelloWorld

Current branch
master

Publish repository
Publish this repository to GitHub

Changes 2

History

README.md

2 changed files

.gitattributes

README.md

Summary (required)

Description

Commit to master

@@ -0,0 +1 @@

1 ++ Helloworld

设置

File Edit View Repository Branch Help

New repository... Ctrl+N

Add local repository... Ctrl+O

Clone repository... Ctrl+Shift+O

Options... Ctrl+,

Exit

Current branch
master

Publish repository
Publish this repository to GitHub

.gitattributes

@@ -0,0 +1,2 @@

1 ++ Auto detect text files and perform LF normalization

2 ++ text=auto

登录，或者一开始就登录，我这里一开始没有登录，随机应变哈

Options

AccountsGitAppearanceAdvanced

GitHub.com

Sign in to your GitHub.com account to access your repositories.

Sign in

Enterprise

If you have a GitHub Enterprise account at work, sign in to it to get access to your repositories.

Sign in

Sign in

Incorrect username or password.

Username or email address

@163.com

Password

.....

Forgot password?

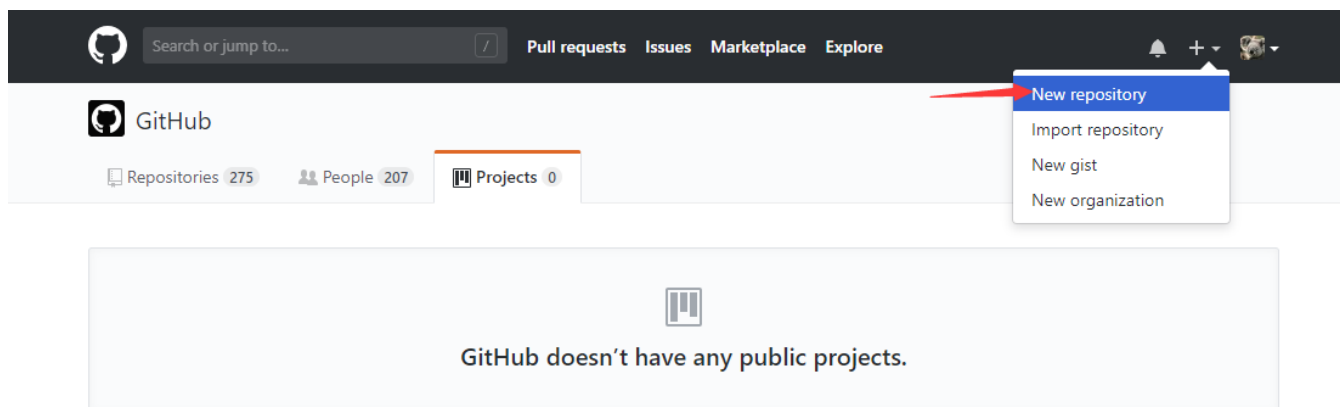
or

Sign in using your browser

Sign inCancel

4、新建一个项目

GitHub 官网新建仓库



Create a new repository

A repository contains all the files for your project, including the revision history.

Owner

 yw-forGit / HelloWorld 

Repository name

Great repository names are short and memorable. Need inspiration? How about **studious-system**.

Description (optional)

第二次在这里向世界问好啦

☒ Public

Anyone can see this repository. You choose who can commit.

该项目共有，则免费，否则收费

☐ Private

You choose who can see and commit to this repository.

☒ Initialize this repository with a README 生成readme文件

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: None

Add a license: None



Create repository

创建完成

yw-forGit / HelloWorld

Watch 0 Star 0 Fork 0

Code

Issues 0

Pull requests 0

Projects 0

Wiki

Insights

Settings

第二次在这里向世界问好啦

Edit

Add topics

1 commit

1 branch

0 releases

1 contributor

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download

yw-forGit Initial commit

Latest commit 10Feb52 36 seconds from now

README.md

Initial commit

just now

README.md

HelloWorld

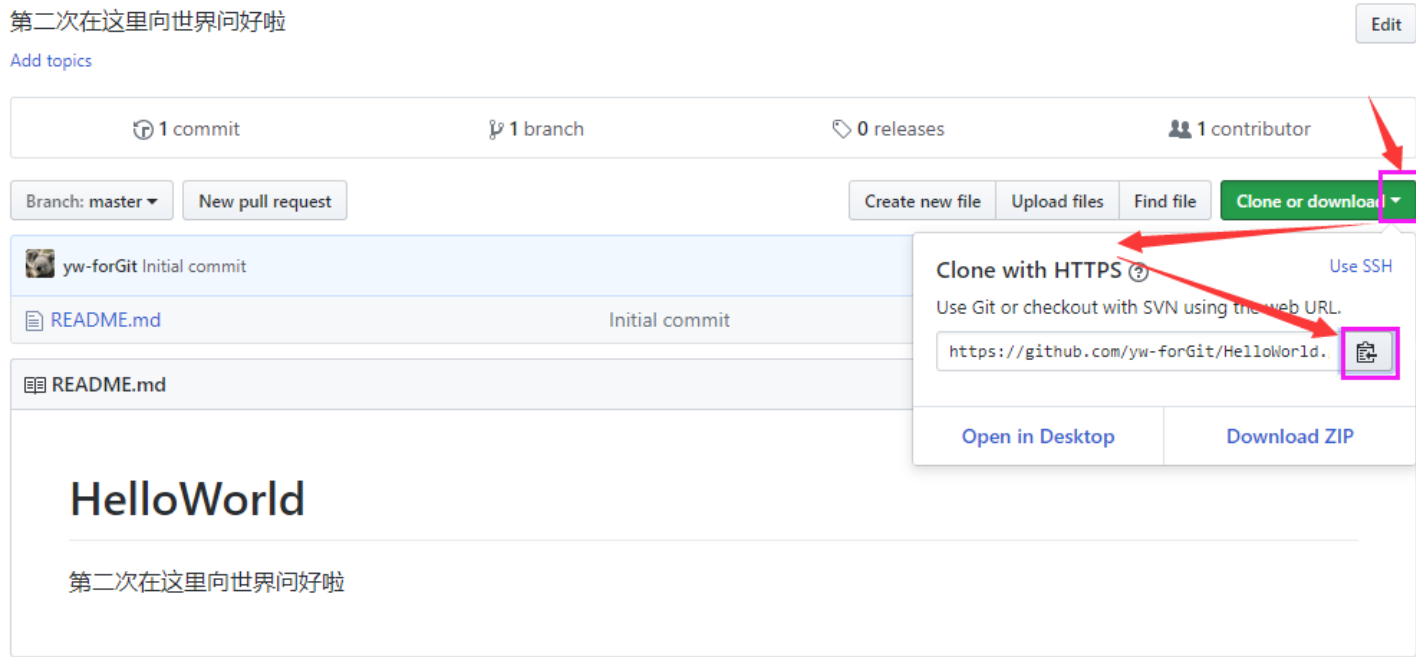
第二次在这里向世界问好啦



5、Linux 环境基本操作

检出项目到本地

复制链接



```
[root@localhost ~]# mkdir git
[root@localhost ~]# cd git
// 复制链接之后, 使用git clone 命令检出到本地
[root@localhost git]# git clone https://github.com/yw-forGit/HelloWorld.git
Cloning into 'HelloWorld'...
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
[root@localhost git]# ll
total 0
drwxr-xr-x. 3 root root 35 Sep  3 22:11 HelloWorld
[root@localhost git]# cd HelloWorld/
[root@localhost HelloWorld]# ll
total 4
-rw-r--r--. 1 root root 50 Sep  3 22:11 README.md
// 这些内容就是在GitHub新建仓库编写的内容
[root@localhost HelloWorld]# cat README.md
# HelloWorld
第二次在这里向世界问好啦
[root@localhost HelloWorld]#
```

第一次提交

新编辑文件

```
// 编辑、新建
[root@localhost HelloWorld]# vi test1
[root@localhost HelloWorld]# cat test1
// 这是我新编辑的内容
今天是x月x日，我吃了一个苹果
// 查看状态
[root@localhost HelloWorld]# git status
// 位于master分支
# On branch master
// 未跟踪的文件是test1文件（提示：使用git add <file>命令可以提交该file的内容）
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#   test1
// 提交为空，但没有可跟踪的文件（提示：使用git add进行跟踪）
nothing added to commit but untracked files present (use "git add" to track)
[root@localhost HelloWorld]#
```

跟踪（git add）

```
// 那就跟踪下吧
[root@localhost HelloWorld]# git add test1
[root@localhost HelloWorld]# git status
# On branch master
// 要提交的变更：新文件test1（提示：使用git reset HEAD <file>命令将变更撤出暂缓区）
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#   new file:   test1
#
[root@localhost HelloWorld]#
```

提交到本地仓库（git commit）

```
// 提交仅仅是提交到本地仓库，GitHub官网还是没有的
[root@localhost HelloWorld]# git commit
```

回车之后，会显示下图，给我们输入描述，按 i 进行编辑，与 vi、vim 编辑一样，编辑完，Esc 退出编辑模式，使用:wq 保存退出即可。


```
1 file changed, 1 insertion(+)
create mode 100644 test1
[root@localhost HelloWorld]#

[root@localhost HelloWorld]# git status
# On branch master
// 你的分支共一个提交。（使用git push命令来发布你的本地提交）
# Your branch is ahead of 'origin/master' by 1 commit.
# (use "git push" to publish your local commits)
#
// 无文件要提交，干净的工作区
nothing to commit, working directory clean
[root@localhost HelloWorld]#
```

push 到 GitHub 官网（git push）

```
// push到GitHub上，如果之前没有配置全局登录信息，则下面会提示输入用户名、密码
[root@localhost HelloWorld]# git push
warning: push.default is unset; its implicit value is changing in
Git 2.0 from 'matching' to 'simple'.
// 若要不再显示本信息并在其默认值改变后维持当前使用习惯，进行如下设置：
To squelch this message and maintain the current behavior after the default changes,
use:

    git config --global push.default matching
// 若要不再显示本信息并从现在开始采用新的使用习惯，设置：
To squelch this message and adopt the new behavior now, use:

    git config --global push.default simple

See 'git help config' and search for 'push.default' for further information.
(the 'simple' mode was introduced in Git 1.7.11. Use the similar mode
'current' instead of 'simple' if you sometimes use older versions of Git)

// 输入GitHub的用户名
Username for 'https://github.com': yw_forgit@163.com
// 密码
Password for 'https://yw_forgit@163.com@github.com':
Counting objects: 4, done.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 348 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/yw-forGit/HelloWorld.git
    10feb52..1bed97c master -> master
// 再查看状态
[root@localhost HelloWorld]# git status
# On branch master
// 无文件要提交，干净的工作区
nothing to commit, working directory clean
```

```
[root@localhost HelloWorld]#
```

去官网看看，已经提交了

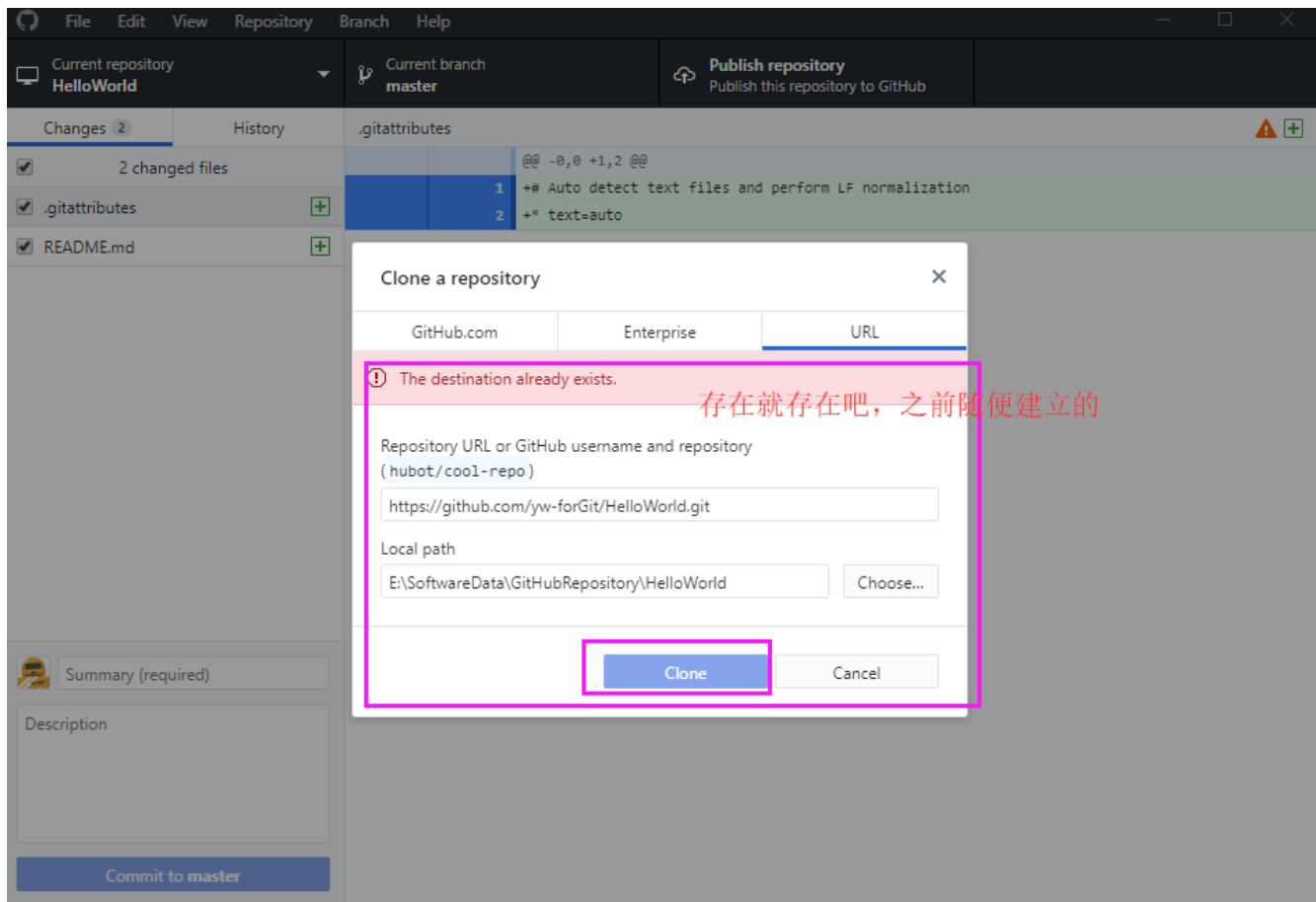
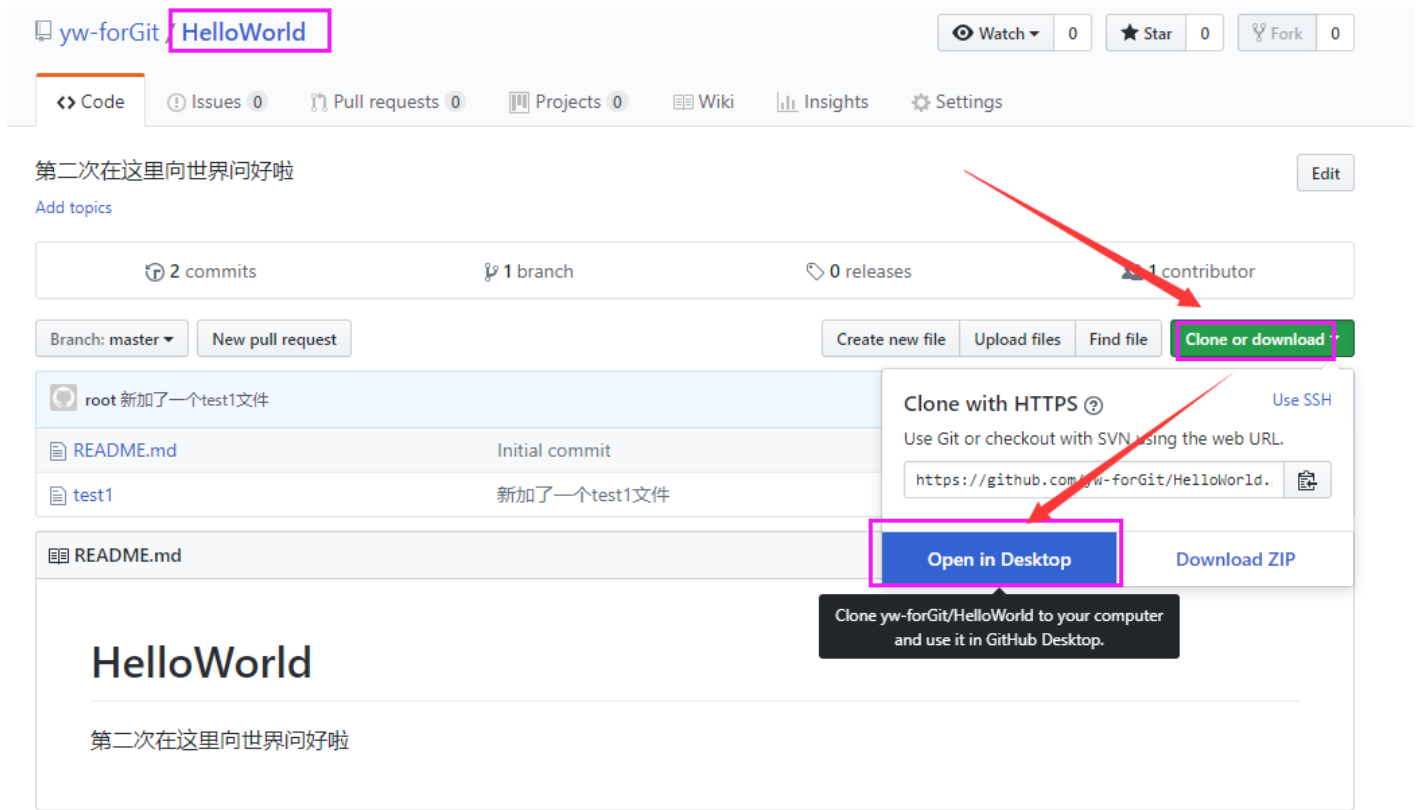
The screenshot shows the GitHub repository page for 'yw-forGit/HelloWorld'. The repository has 2 commits, 1 branch, 0 releases, and 1 contributor. The commit history shows two commits: 'Initial commit' (an hour ago) and 'test1' (20 minutes ago). The 'test1' commit is highlighted with a pink box. The file 'test1' is also highlighted with a pink box. The README.md file is visible, containing the text 'HelloWorld' and '第二次在这里向世界问好啦'.

The screenshot shows the GitHub file view for the 'test1' file. The file is located in the 'HelloWorld' repository. The file content is '1 今天是x月x日，我吃了一个苹果'. The file is highlighted with a pink box.

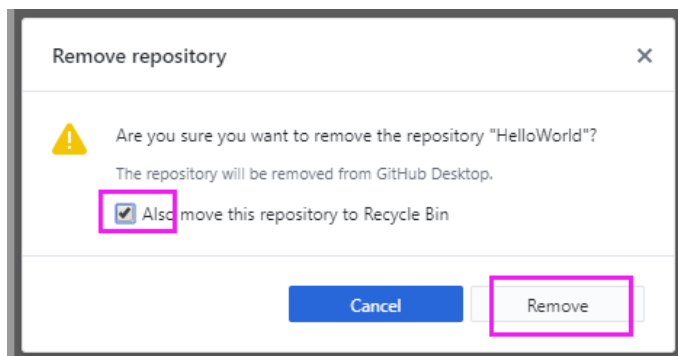
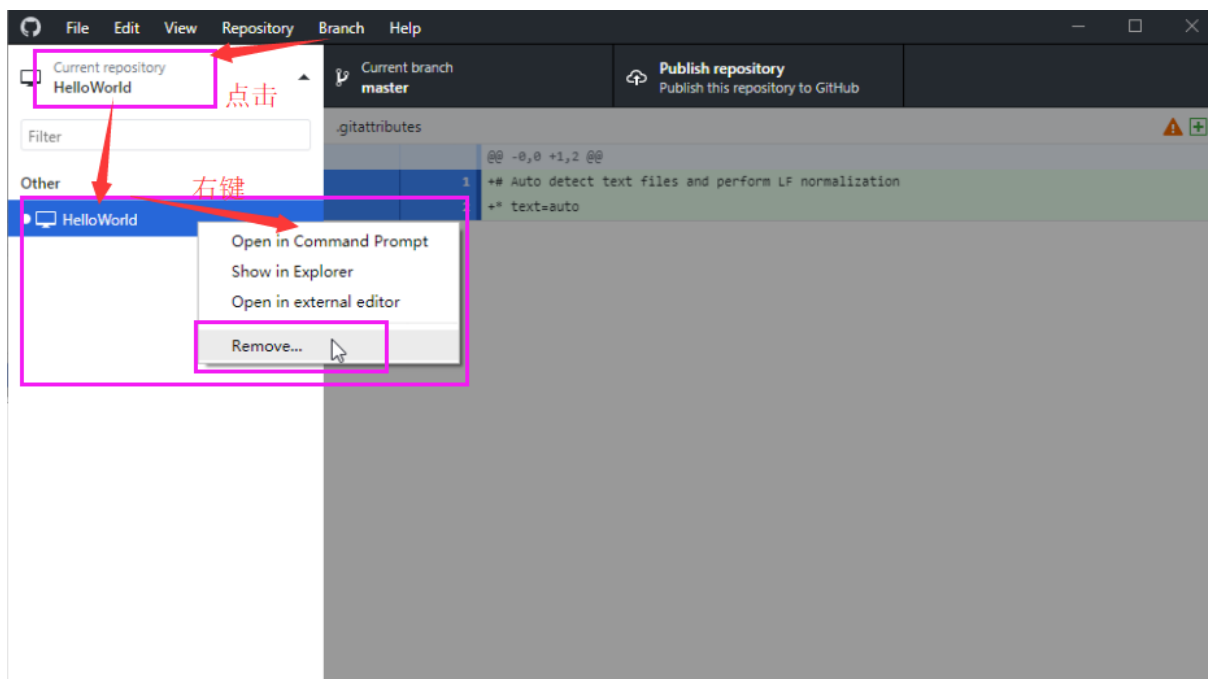
6、Windows 环境基本操作

检出项目到本地

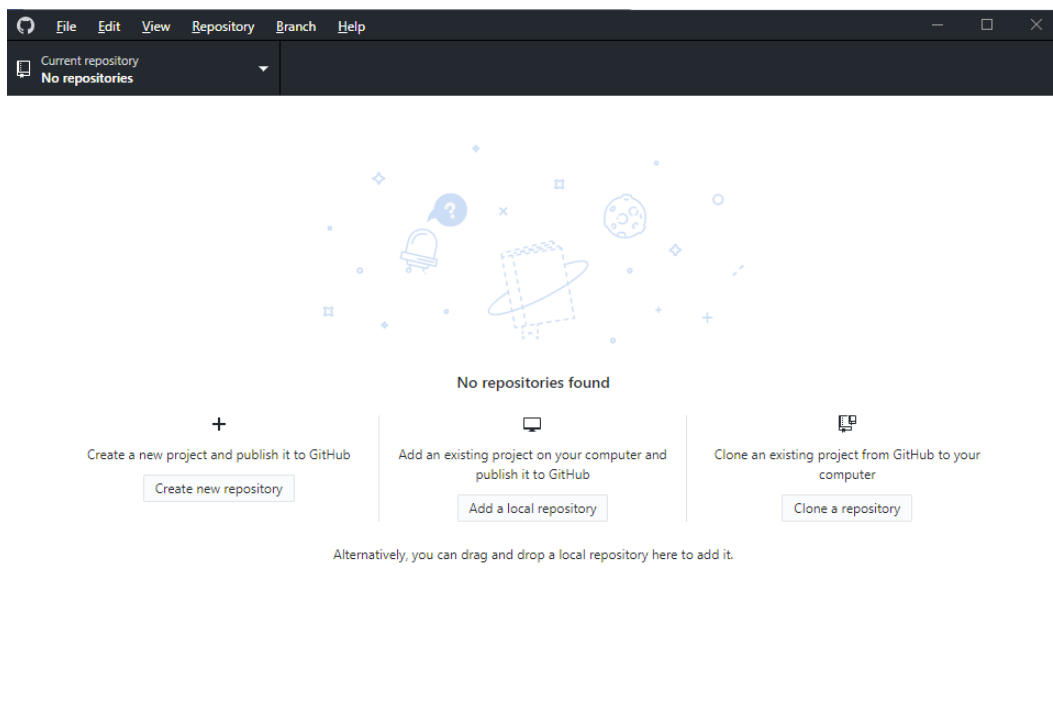
打开官网，进行克隆，然后打开客户端



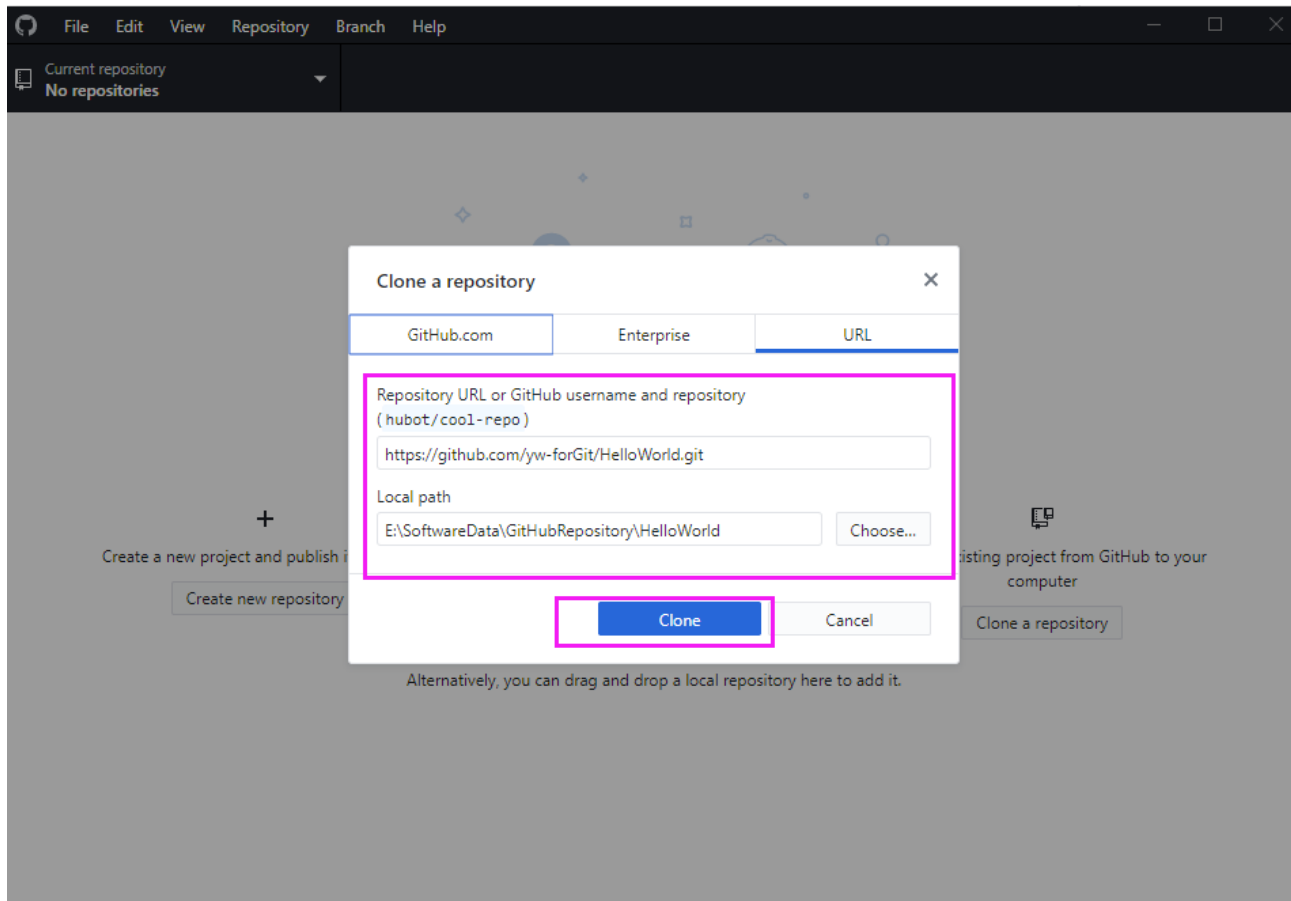
那就 remove 掉吧



干干净净的



回到官网，重新克隆

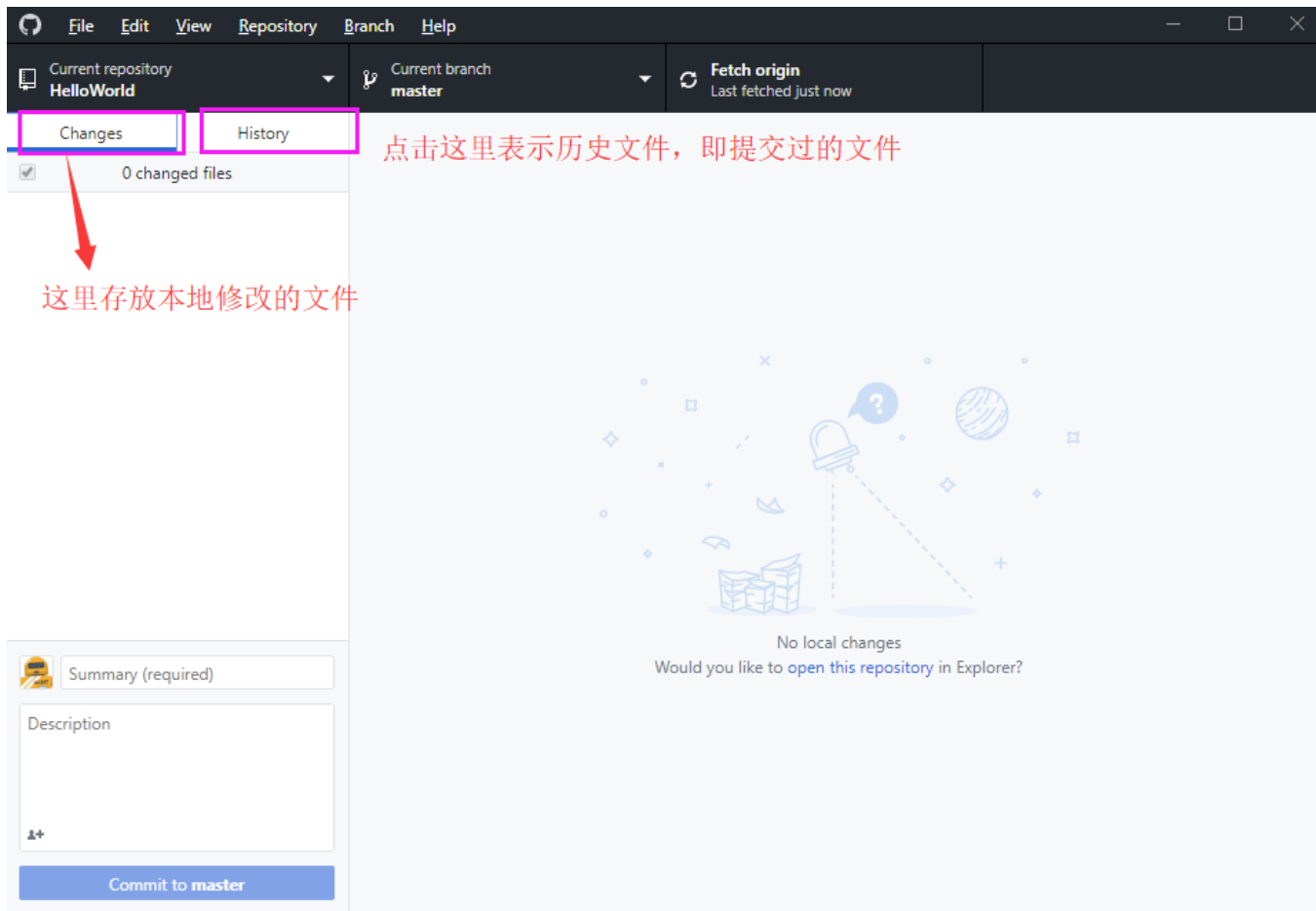


Cloning HelloWorld

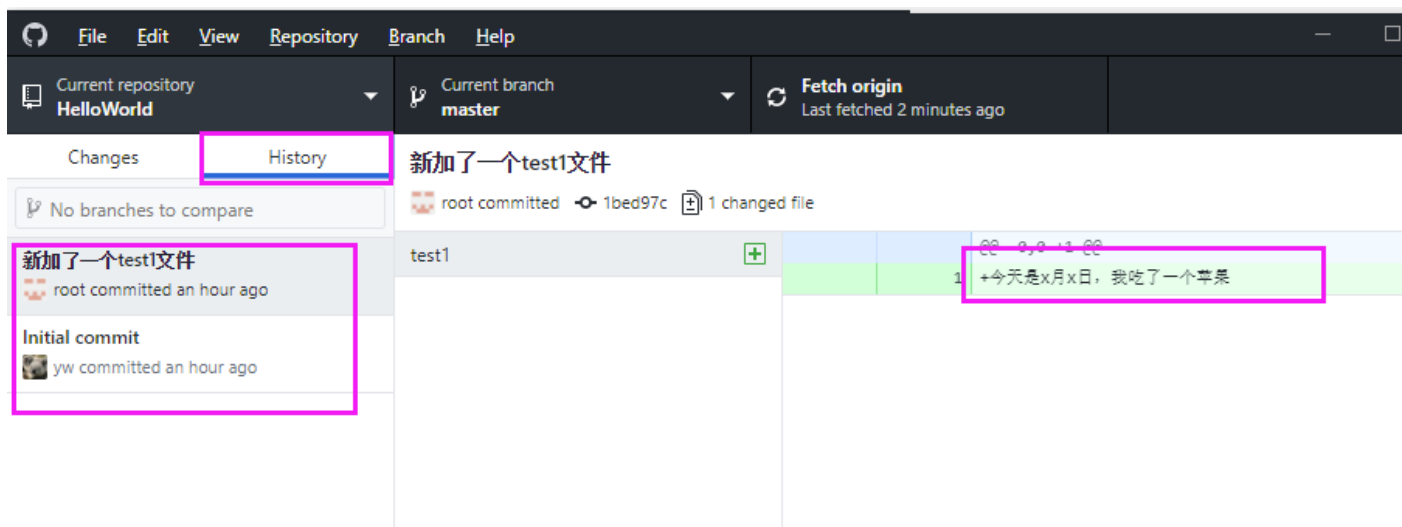


remote: Total 6 (delta 0), reused 3 (delta 0), pack-reused 0

克隆完成，空白不要慌。

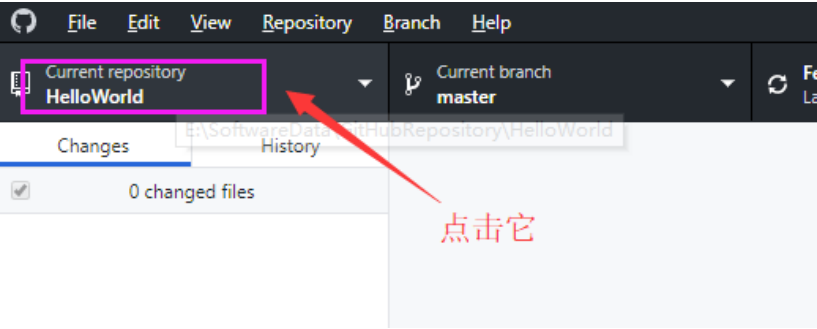


点击“History”，可以看到提交过的文件

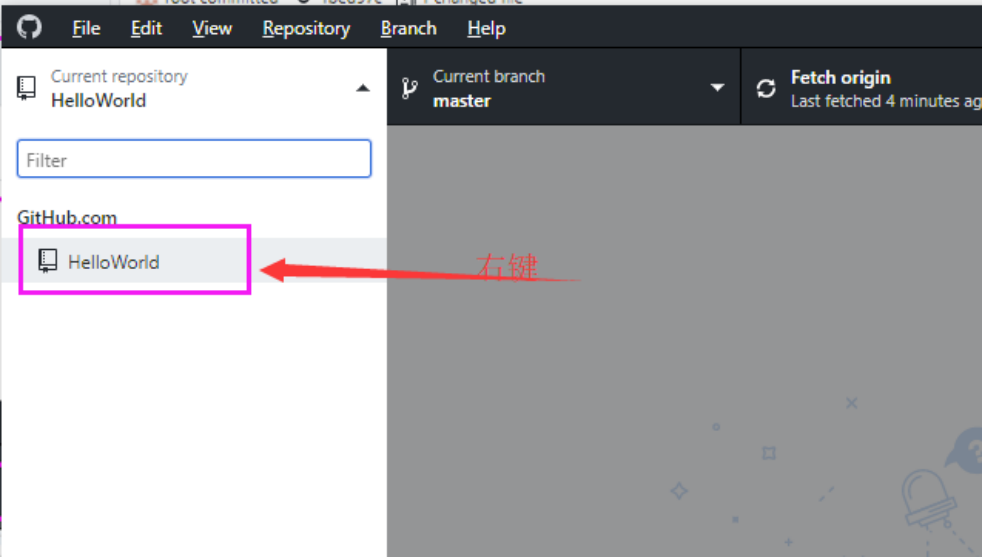


第一次提交

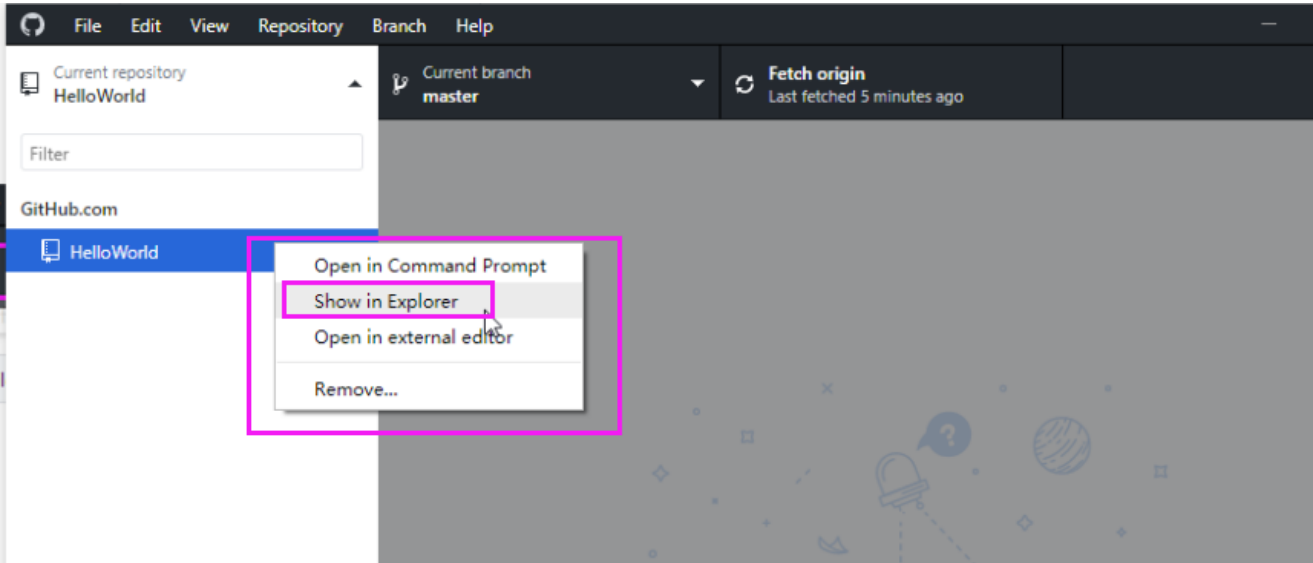
编辑文件

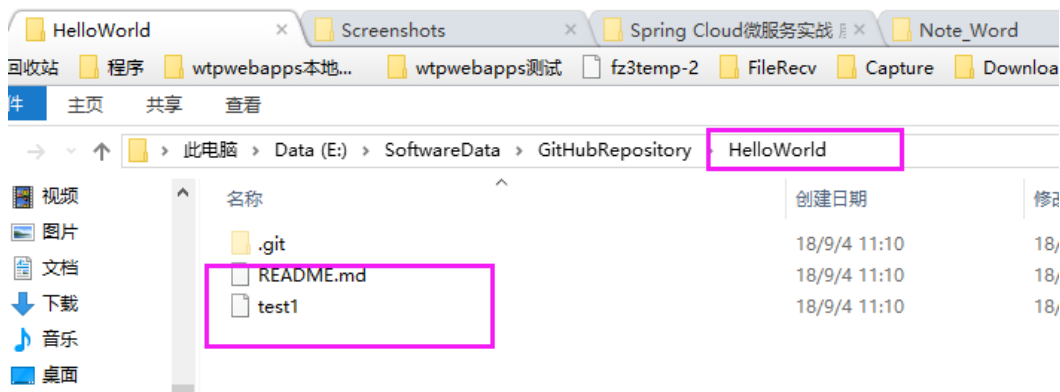


就会出现下图

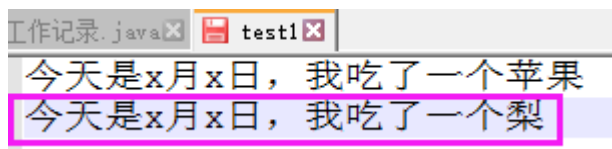


右键之后，出现下图，选择 Show in Explorer，

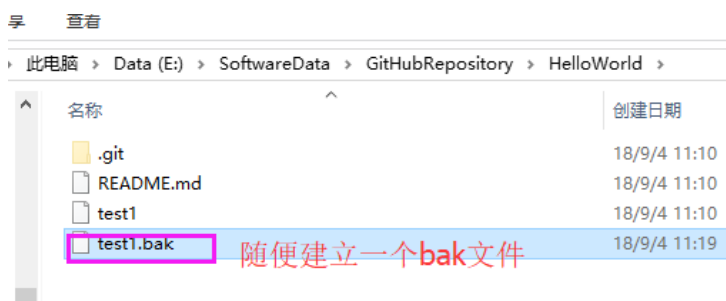




用文本编辑器打开，新添加一行文字

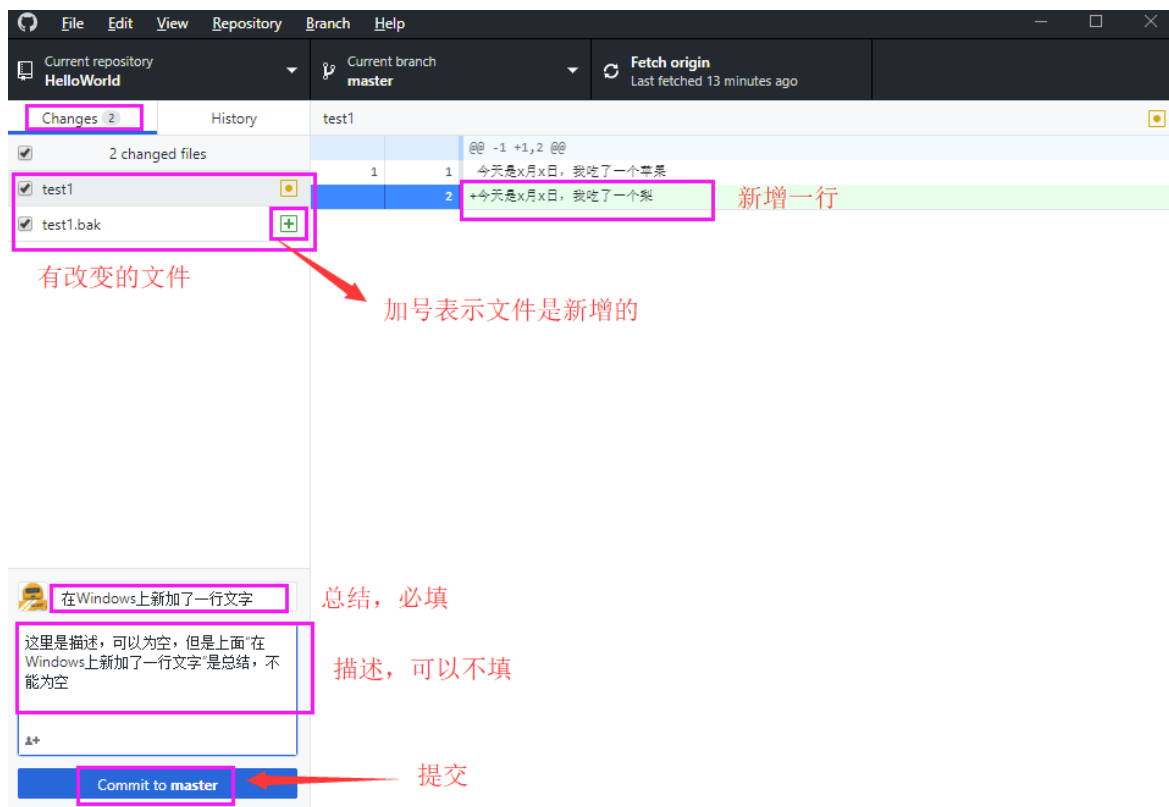


再随便建立一个 bak 文件，后面用到



提交到本地

回到 GitHub 界面，最后选择提交



哎呀，报错了。Commit failed - exit code 128 received, with output: '*** Please tell me who you are.

Error

❗

Commit failed - exit code 128 received, with output: '*** Please tell me who you are.

Run

```
git config --global user.email "you@example.com"
git config --global user.name "Your Name"
```

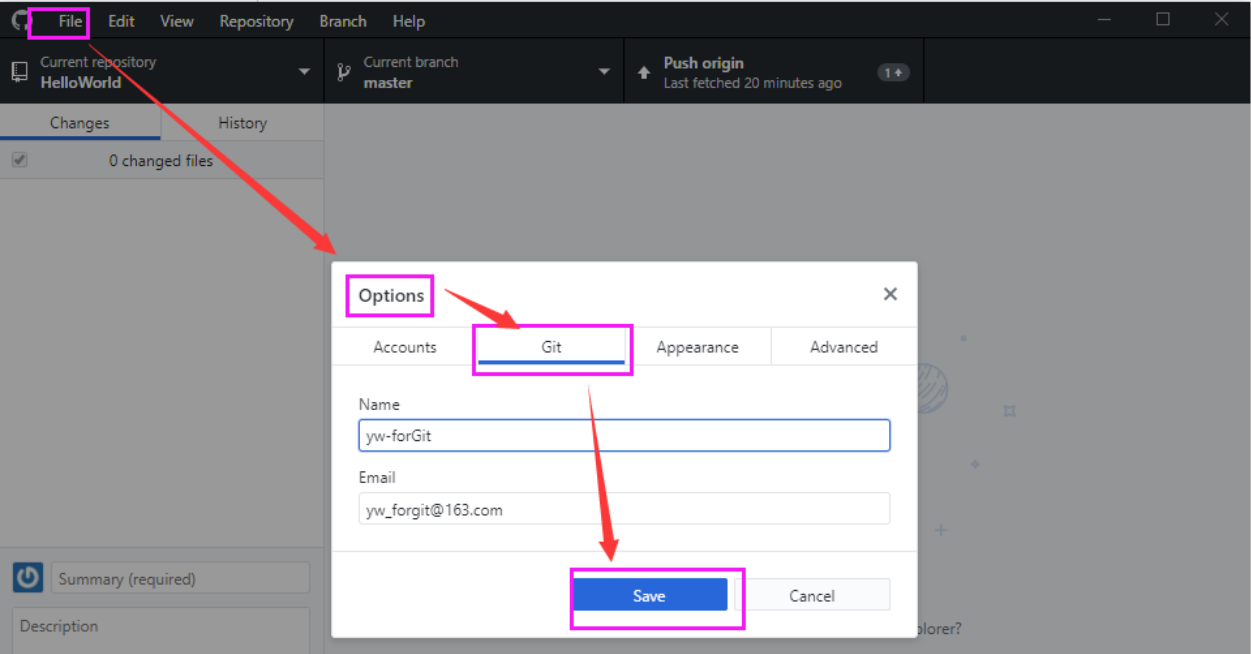
to set your account's default identity.

Omit --global to set the identity only in this repository.

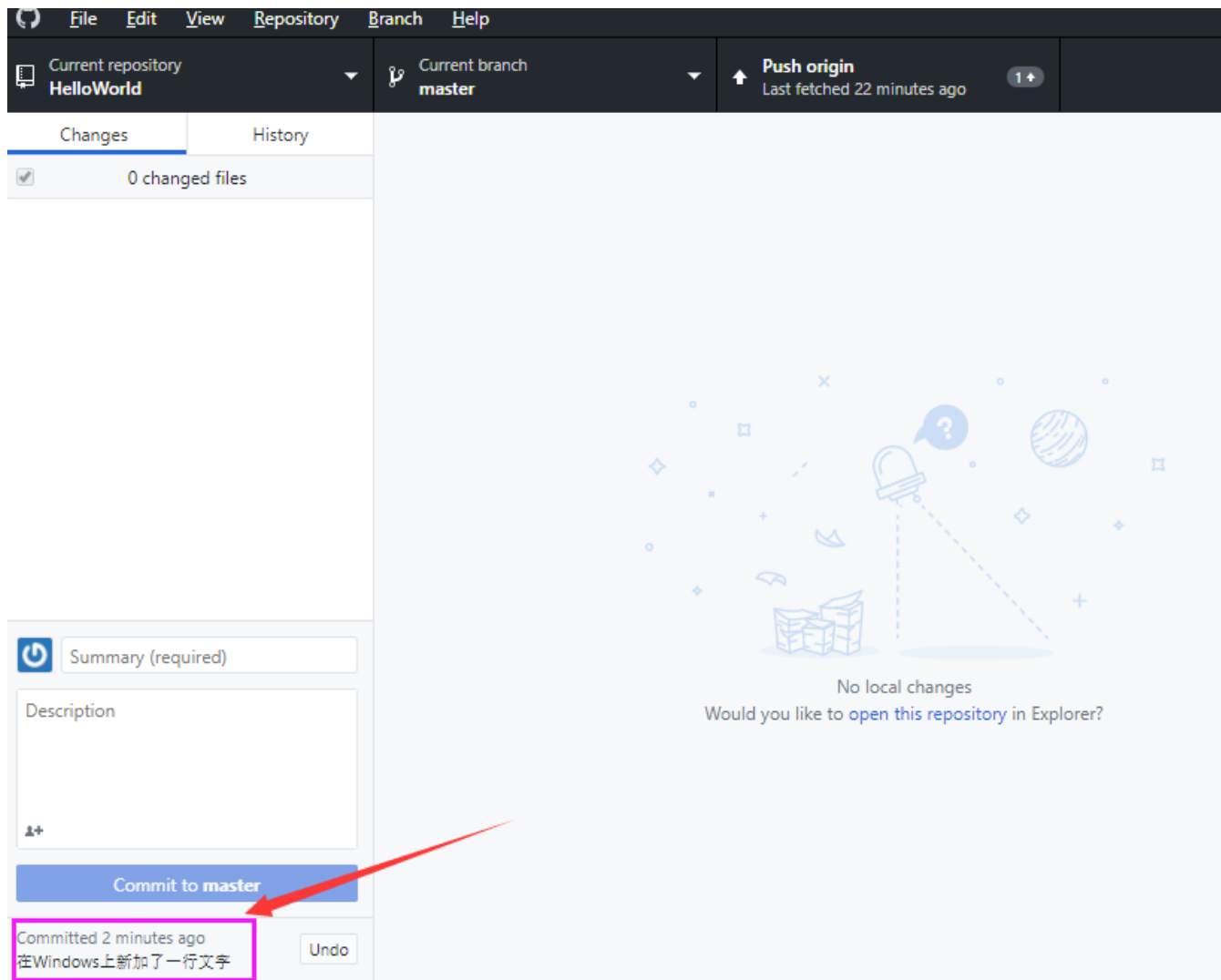
```
fatal: unable to auto-detect email address (got 'WIN@DESKTOP-HEKHU8H.(none)')
```

Close

我去，去网站看到这样操作一遍就好了，但是我什么都没有改，Name、Email 都已经有了的。

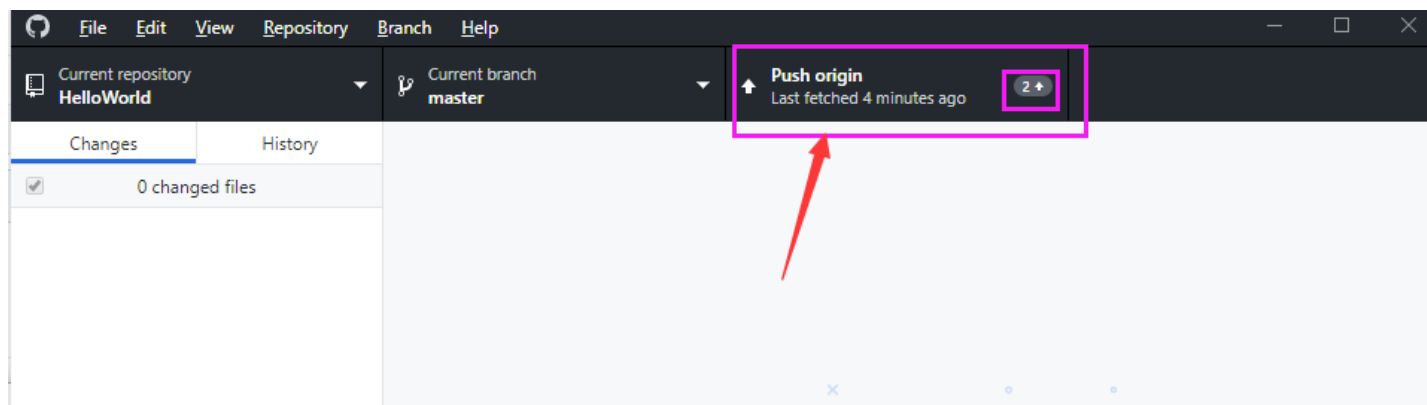


然后再次提交，就好了。但仅仅是提交到本地

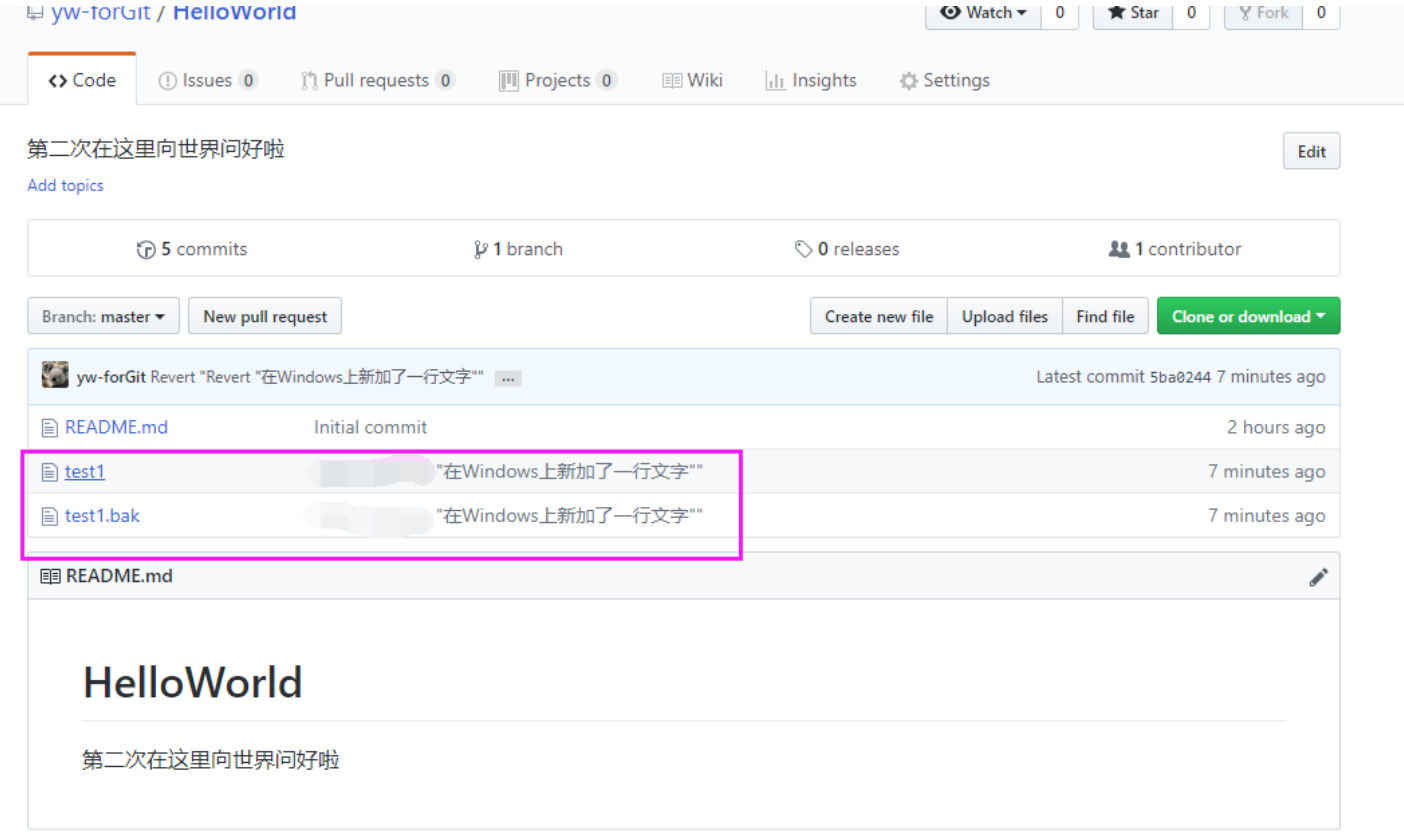


push 到官网

点击下图的标注部分，即可

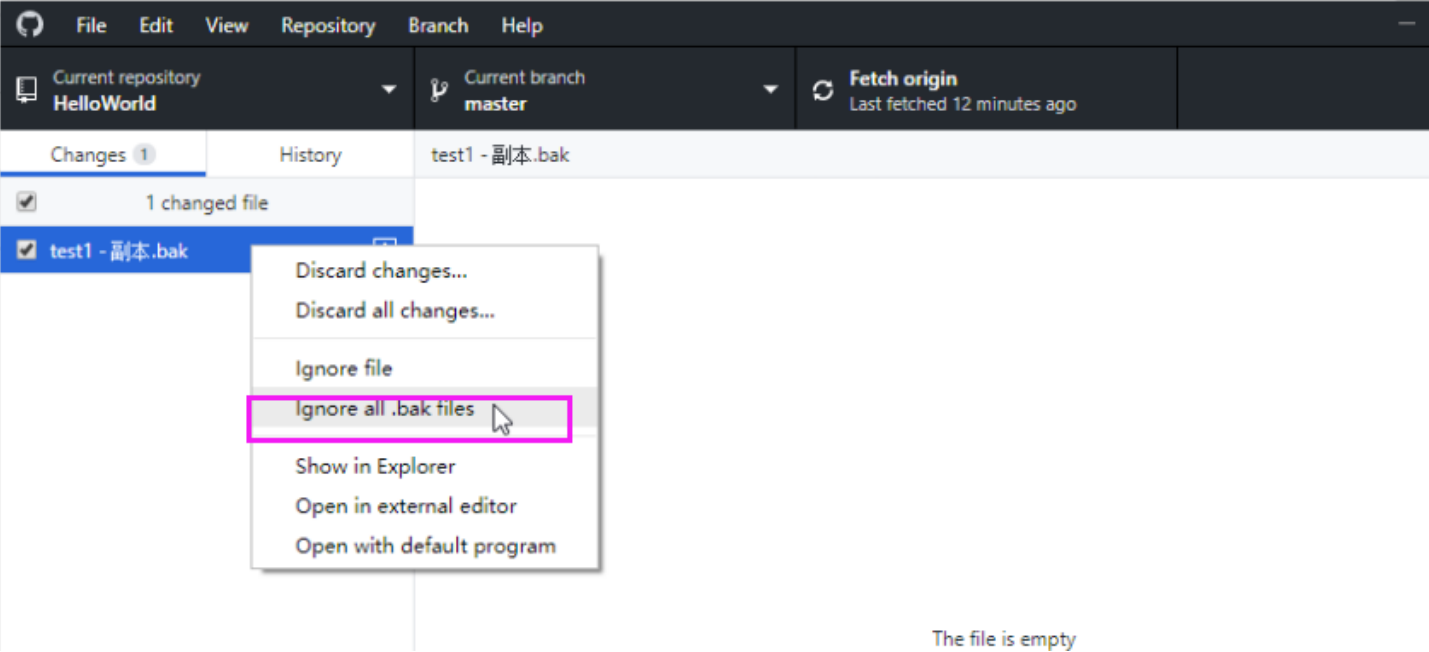


看下官网，哎呀，bak 文件其实不用提交的，一般都是编辑器自动生成的备份文件。

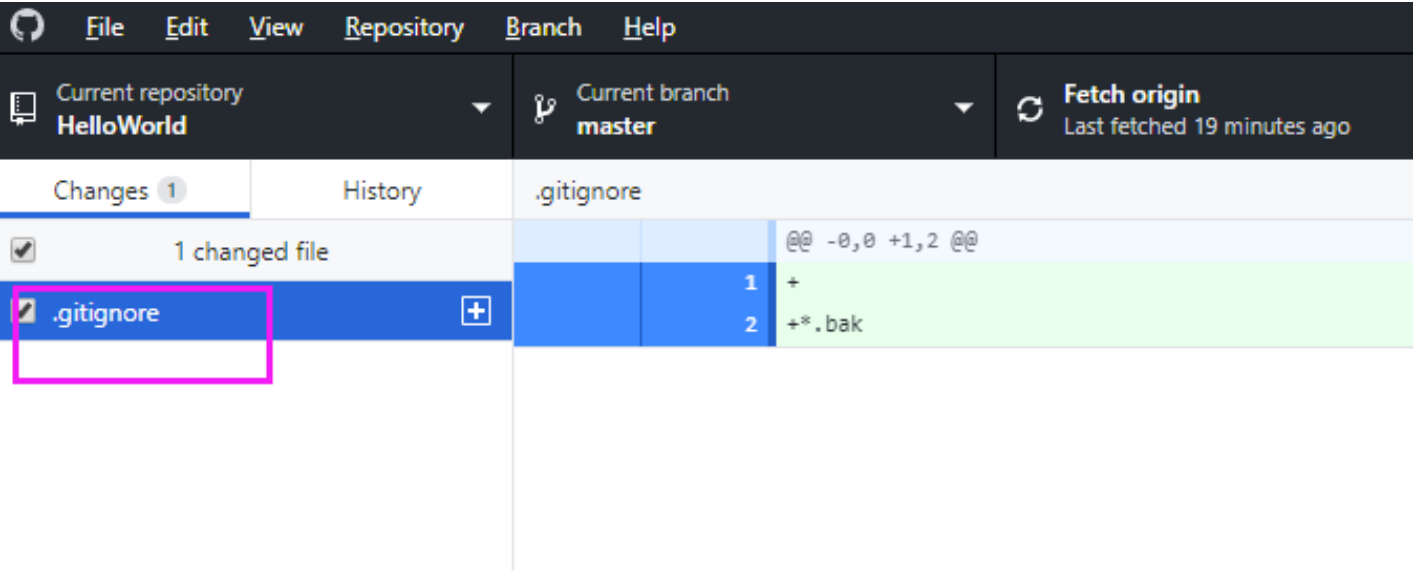


忽略文件

忽略固定后缀文件

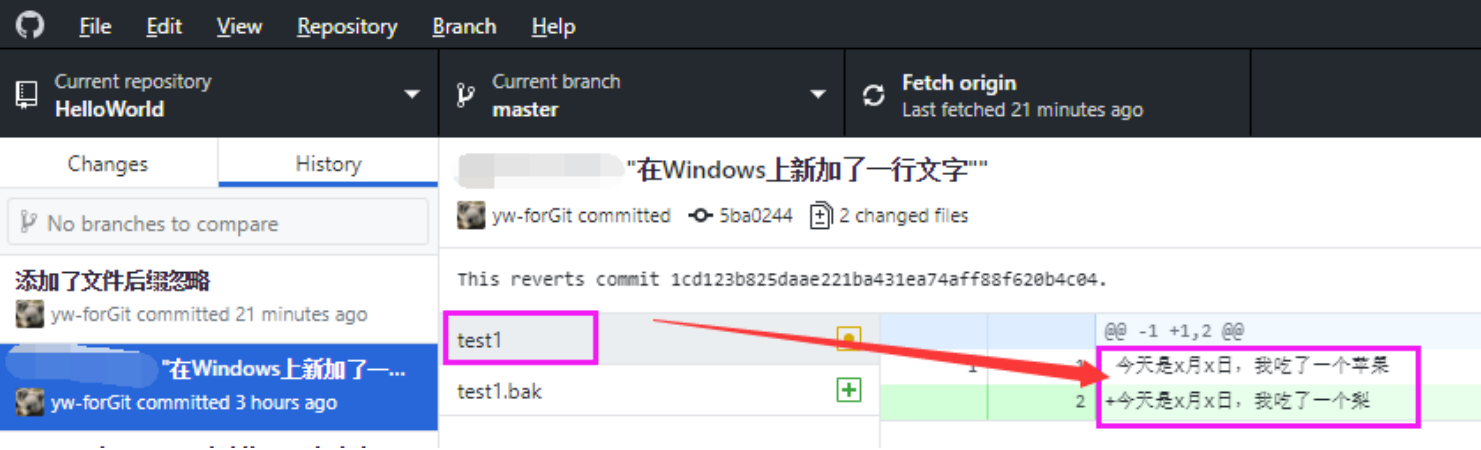


然后就会出现这个文件，该文件就是记录哪些文件被忽略的，建议提交上去，这样一个团队的忽略后缀都是一样的。



7、解决冲突

本来在第 6 节的时候，在 Windows 时，test1 被改成了下图的样子。



但是在第 5 节，Linux 本地仓库还是这样的版本

```
[root@localhost HelloWorld]# cat test1
今天是 x 月 x 日，我吃了一个苹果
```

本来应该是先 git pull 拉取一下代码再修改的，但是这里故意制造冲突。下面是 Linux 的操作

```
[root@localhost HelloWorld]# pwd
/root/git/HelloWorld
[root@localhost HelloWorld]# ll
total 8
-rw-r--r--. 1 root root 50 Sep  3 22:11 README.md
-rw-r--r--. 1 root root 42 Sep  3 22:15 test1
// Linux本地仓库跟GitHub官网的不一样了
[root@localhost HelloWorld]# cat test1
今天是x月x日，我吃了一个苹果
[root@localhost HelloWorld]# vi test1
// 可以发现增加了一句话
[root@localhost HelloWorld]# cat test1
今天是x月x日，我吃了一个苹果
我是第一个改了这里
```

```
[root@localhost HelloWorld]# git status
# On branch master
#
# Changes not staged for commit:
#
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       modified:   test1
#
no changes added to commit (use "git add" and/or "git commit -a")
```

```
[root@localhost HelloWorld]# git add test1
```

```
[root@localhost HelloWorld]# git commit
```

```
[master 2794d53] 我是第一个修改的人提交了代码
```

```
Committer: root <root@localhost.localdomain>
```

Your name and email address were configured automatically based on your username and hostname. Please check that they are accurate.

You can suppress **this** message by setting them explicitly:

```
git config --global user.name "Your Name"
```

```
git config --global user.email you@example.com
```

After doing **this**, you may fix the identity used **for this** commit with:

```
git commit --amend --reset-author
```

```
1 file changed, 1 insertion(+)
```

```
[root@localhost HelloWorld]# git push
```

warning: push.default is unset; its implicit value is changing in

Git 2.0 from 'matching' to 'simple'. To squelch **this** message

and maintain the current behavior after the **default** changes, use:

```
git config --global push.default matching
```

To squelch **this** message and adopt the **new** behavior now, use:

```
git config --global push.default simple
```

See 'git help config' and search **for** 'push.default' **for** further information.

(the 'simple' mode was introduced in Git 1.7.11. Use the similar mode

'current' instead of 'simple' **if** you sometimes use older versions of Git)

Username **for** 'https://github.com': yw_forgit@163.com

Password **for** 'https://yw_forgit@163.com@github.com':

To <https://github.com/yw-forGit/HelloWorld.git>

```
! [rejected]        master -> master (fetch first)
```

```
error: failed to push some refs to 'https://github.com/yw-forGit/HelloWorld.git'
```

// 提示：更新被拒绝，因为远程版本库包含你本地尚不存在的提交

hint: Updates were rejected because the remote contains work that you **do**

hint: not have locally. This is usually caused by another repository pushing

// 你要先合并远程变更，再提交

hint: to the same ref. You may want to first merge the remote changes (e.g.,

hint: 'git pull') before pushing again.

hint: See the 'Note about fast-forwards' in 'git push --help' **for** details.

```
// 那我就去远程官网拉取一下代码
[root@localhost HelloWorld]# git pull

remote: Counting objects: 9, done.

remote: Compressing objects: 100% (7/7), done.

remote: Total 9 (delta 1), reused 8 (delta 0), pack-reused 0

Unpacking objects: 100% (9/9), done.

From https://github.com/yw-forGit/HelloWorld
   1bed97c..2d24cee master    -> origin/master

Auto-merging test1
CONFLICT (content): Merge conflict in test1

Automatic merge failed; fix conflicts and then commit the result.

[root@localhost HelloWorld]# ll

total 8

-rw-r--r--. 1 root root  50 Sep  3 22:11 README.md
-rw-r--r--. 1 root root 179 Sep  4 02:13 test1
-rw-r--r--. 1 root root   0 Sep  4 02:13 test1.bak

[root@localhost HelloWorld]# cat test1
今天是x月x日，我吃了一个苹果
// 这段是我Linux自己修改的
<<<<<<< HEAD
我是第一个改了这里
=====
// 这段是我Windows那家伙修改的
今天是x月x日，我吃了一个梨
>>>>>>> 2d24cee58d065f5163ff79c7a652cda4b1ffe550

[root@localhost HelloWorld]# vi test1
[root@localhost HelloWorld]# vi test1
// 合并，都保留
[root@localhost HelloWorld]# cat test1
今天是x月x日，我吃了一个苹果
我是第一个改了这里
今天是x月x日，我吃了一个梨
[root@localhost HelloWorld]# git status

# On branch master

# Your branch and 'origin/master' have diverged,
# and have 1 and 4 different commits each, respectively.
#   (use "git pull" to merge the remote branch into yours)
#

# You have unmerged paths.
#   (fix conflicts and run "git commit")
#

# Changes to be committed:
#
#   new file:   .gitignore
#   new file:   test1.bak
#
# Unmerged paths:
#   (use "git add <file>..." to mark resolution)
#
#   both modified:    test1
#
```

```
[root@localhost HelloWorld]# git add test1
[root@localhost HelloWorld]# git commit
```

弹框下图，然后我在首行写了注释

```
1 虚拟机CentOS7 x 2 虚拟机CentOS7 x +
我修改了冲突
Merge branch 'master' of https://github.com/yw-forGit/HelloWorld

Conflicts:
    test1
#
# It looks like you may be committing a merge.
# If this is not correct, please remove the file
#   .git/MERGE_HEAD
# and try again.

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# Committer: root <root@localhost.localdomain>
#
# On branch master
# Your branch and 'origin/master' have diverged,
# and have 1 and 4 different commits each, respectively.
#   (use "git pull" to merge the remote branch into yours)
#
# All conflicts fixed but you are still merging.
#   (use "git commit" to conclude merge)
#
# Changes to be committed:
#
#       new file:   .gitignore
#       modified:   test1
#       new file:   test1.bak
#
~
~
~
```

```
[master 0c79dee] 我修改了冲突 Merge branch 'master' of https://github.com/yw-forGit/HelloWorld
```

```
Committer: root <root@localhost.localdomain>
```

Your name and email address were configured automatically based on your username and hostname. Please check that they are accurate. You can suppress **this** message by setting them explicitly:

```
git config --global user.name "Your Name"
git config --global user.email you@example.com
```

After doing **this**, you may fix the identity used **for this** commit with:

```
git commit --amend --reset-author
```

```
[root@localhost HelloWorld]# git push
```

warning: push.default is unset; its implicit value is changing in Git 2.0 from 'matching' to 'simple'. To squelch **this** message and maintain the current behavior after the **default** changes, use:

```
git config --global push.default matching
```

To squelch **this** message and adopt the **new** behavior now, use:

```
git config --global push.default simple
```

See 'git help config' and search **for** 'push.default' **for** further information.

(the 'simple' mode was introduced in Git 1.7.11. Use the similar mode 'current' instead of 'simple' **if** you sometimes use older versions of Git)

Username **for** 'https://github.com': yw_forgit@163.com

Password **for** 'https://yw_forgit@163.com@github.com':

Counting objects: 10, done.

Compressing objects: 100% (6/6), done.

Writing objects: 100% (6/6), 750 bytes | 0 bytes/s, done.

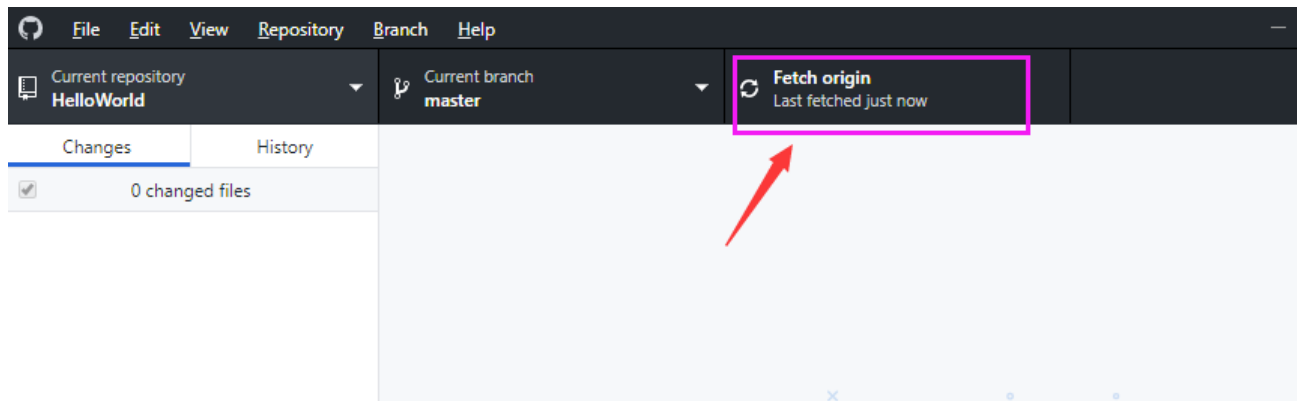
Total 6 (delta 2), reused 0 (delta 0)

remote: Resolving deltas: 100% (2/2), completed with 1 local object.

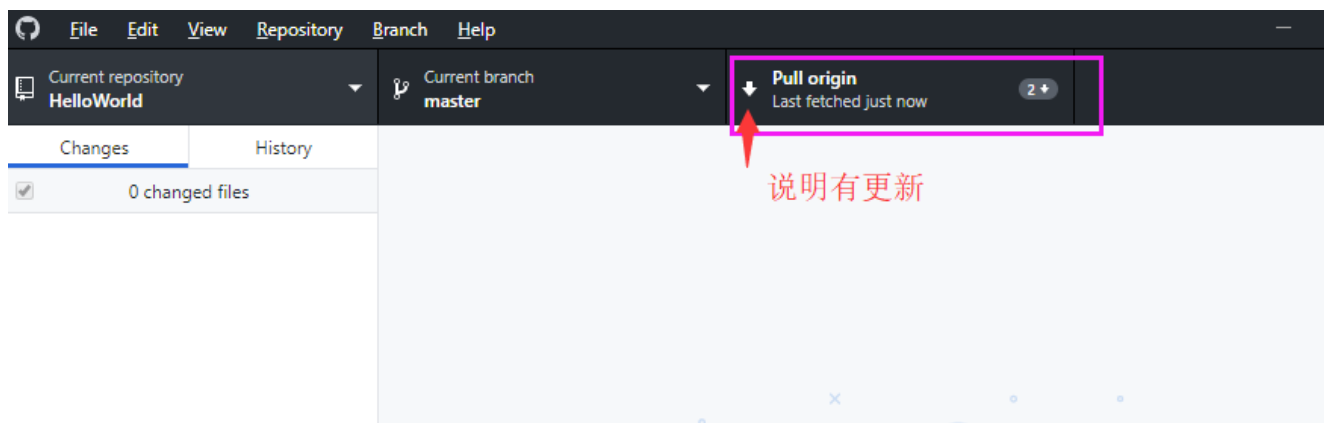
To <https://github.com/yw-forGit/HelloWorld.git>

2d24cee..0c79dee master -> master

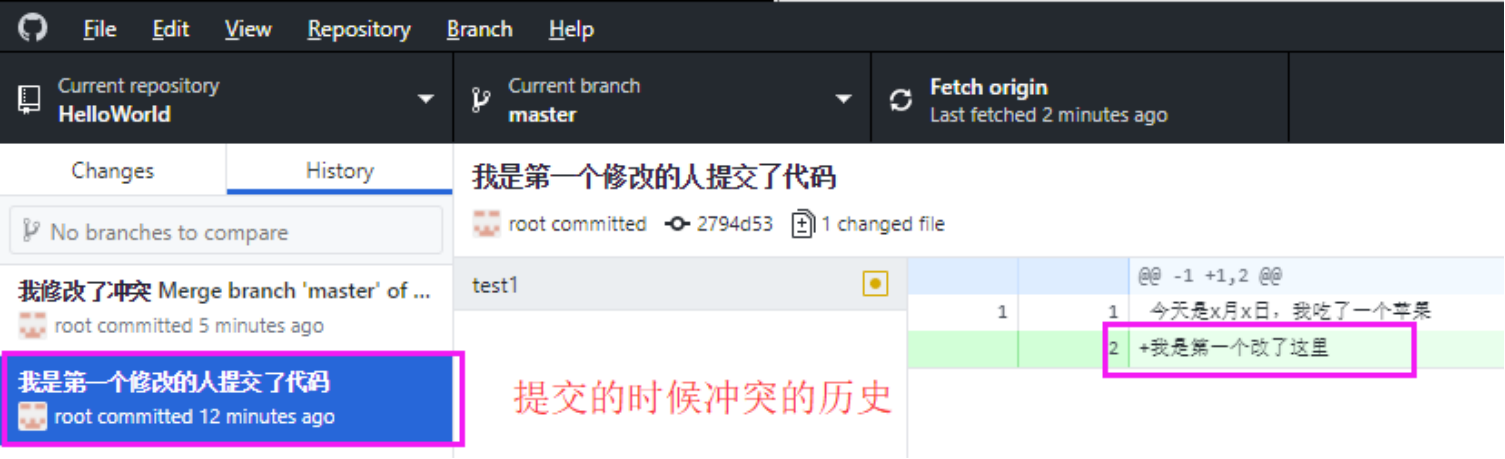
回到 Windows 客户端，点击同步一下，获取 GitHub 官网的东西



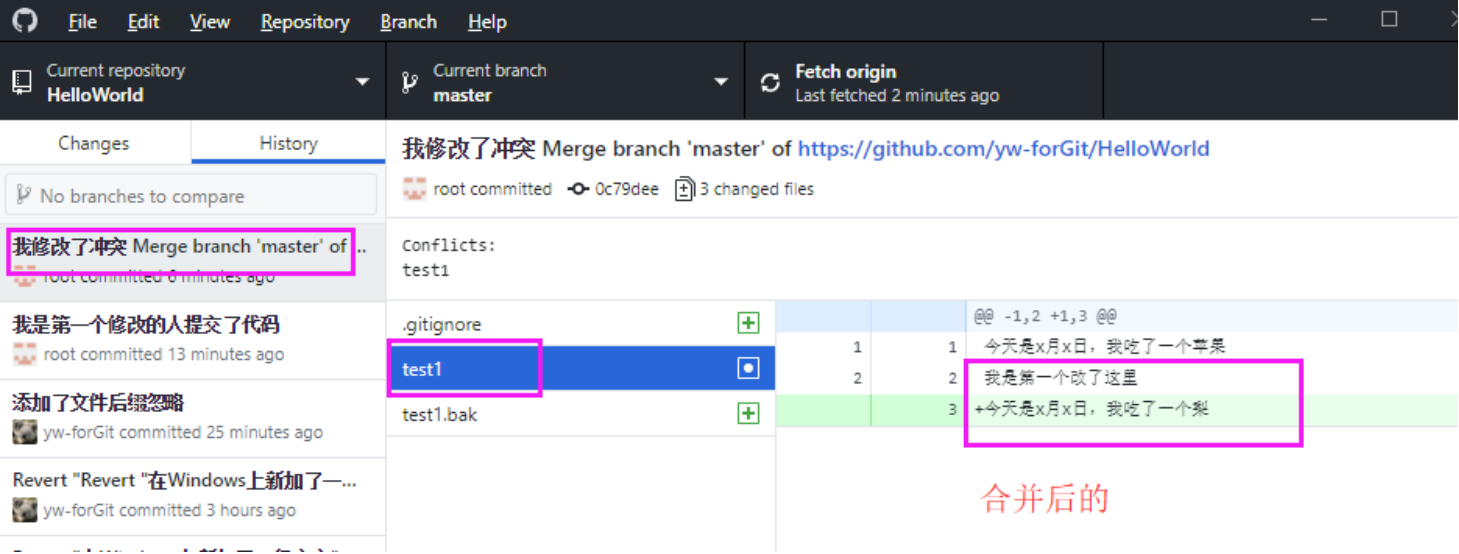
拉取下来



看下冲突的历史也在这里



在 Windows 客户端也看到合并后的最新结果



Windows 解决冲突也是一样的。

8、回到过去、去到未来

查看一下更新的日志

```
[root@localhost HelloWorld]# git log
commit 0c79dee3e450ba9631c8ba4049394bf16bb3e6c3
Merge: 2794d53 2d24cee
Author: root <root@localhost.localdomain>
Date: Tue Sep 4 02:17:28 2018 -0400

    我修改了冲突
    Merge branch 'master' of https://github.com/yw-forGit/HelloWorld

Conflicts:
    test1

commit 2794d5368a303fe9eadb865178afbc6ce223d087
Author: root <root@localhost.localdomain>
Date: Tue Sep 4 02:09:57 2018 -0400
```

我是第一个修改的人提交了代码

```
commit 2d24cee58d065f5163ff79c7a652cda4b1ffe550
Author: yw-forGit <yw_forgit@163.com>
Date: Tue Sep 4 13:58:05 2018 +0800
```

添加了文件后缀忽略

添加了文件后缀忽略

```
commit 5ba024458da448a4ab232900689b597563b7b377
Author: yw-forGit <yw_forgit@163.com>
Date: Tue Sep 4 11:37:55 2018 +0800
```

Revert "Revert "在Windows上新加了一行文字""

This reverts commit 1cd123b825daae221ba431ea74aff88f620b4c04.

```
commit 1cd123b825daae221ba431ea74aff88f620b4c04
Author: yw-forGit <yw_forgit@163.com>
Date: Tue Sep 4 11:34:52 2018 +0800
```

Revert "在Windows上新加了一行文字"

This reverts commit 5bcf406a9589a6478bddef388dfc37b8d05e9e4b.

```
commit 5bcf406a9589a6478bddef388dfc37b8d05e9e4b
Author: yw-forGit <yw_forgit@163.com>
Date: Tue Sep 4 11:30:19 2018 +0800
```

在Windows上新加了一行文字

这里是描述，可以为空，但是上面“在Windows上新加了一行文字”是总结，不能为空

```
commit 1bed97cfd9c24fd0cfbbcbba2499dc96336ea955
Author: root <root@localhost.localdomain>
Date: Mon Sep 3 22:25:16 2018 -0400
```

新加了一个test1文件

```
commit 10feb52d94290aab5727cb6ffff53304a53fd48f
Author: yw <yw_forgit@163.com>
Date: Tue Sep 4 09:52:46 2018 +0800
```

Initial commit

如果卡在这里不知道怎么出来，就按“:q”就能出来了

```
commit 10feb52d94290aab5727cb6ffff53304a53fd48f
Author: yw <yw_forgit@163.com>
Date: Tue Sep 4 09:52:46 2018 +0800

Initial commit

(END)
```

```
[root@localhost HelloWorld]#
// 当前最新版本
[root@localhost HelloWorld]# cat test1
今天是x月x日，我吃了一个苹果
我是第一个改了这里
今天是x月x日，我吃了一个梨
// 回到过去。随便恢复到一个版本，要用版本id来恢复
[root@localhost HelloWorld]# git reset --hard
2d24cee58d065f5163ff79c7a652cda4b1ffe550
HEAD is now at 2d24cee 添加了文件后缀忽略
[root@localhost HelloWorld]# cat test1
今天是x月x日，我吃了一个苹果
今天是x月x日，我吃了一个梨
// 去到未来。同理，随便恢复到一个版本，要用版本id来恢复
[root@localhost HelloWorld]# git reset --hard
0c79dee3e450ba9631c8ba4049394bf16bb3e6c3
HEAD is now at 0c79dee 我修改了冲突
[root@localhost HelloWorld]# cat test1
今天是x月x日，我吃了一个苹果
我是第一个改了这里
今天是x月x日，我吃了一个梨
[root@localhost HelloWorld]#
```

9、建立里程碑

第二次在这里向世界问好啦

Edit

[Add topics](#)

8 commits

1 branch

0 releases

2 contributors

Branch: master

[New pull request](#)[Create new file](#)[Upload files](#)[Find file](#)[Clone or download](#)

root 我修改了冲突 ...

Latest commit 0c79dee 35 minutes ago

.gitignore	添加了文件后忽略	an hour ago
README.md	Initial commit	5 hours ago
test1	我修改了冲突	35 minutes ago
test1.bak	Revert "Revert "在Windows上新加了一行文字""	3 hours ago

[README.md](#)

Releases

Tags



There aren't any releases here

Releases are powered by [tagging specific points of history](#) in a repository.

They're great for marking release points like `v1.0`.

[Create a new release](#)

<> Code

Issues 0

Pull requests 0

Projects 0

Wiki

Insights

Settings

ReleasesTags

1.0 @ Target: master

Excellent! This tag will be created from the target when you publish this release.

Release title

WritePreview

1.0版本

Attach files by dragging & dropping, selecting them, or pasting from the clipboard.

Attach binaries by dropping them here or selecting them.

☐ This is a pre-release
We'll point out that this release is identified as non-production ready.

Publish releaseSave draft

Tagging suggestions

It's common practice to prefix your version names with the letter v. Some good tag names might be v1.0 or v2.3.4.

If the tag isn't meant for production use, add a pre-release version after the version name. Some good pre-release versions might be v0.2-alpha or v5.9-beta.3.

Semantic versioning

If you're new to releasing software, we highly recommend reading about semantic versioning.

yw-forGit / HelloWorld

Watch 0

Star 0

Fork 0

<> Code

Issues 0

Pull requests 0

Projects 0

Wiki

Insights

Settings

ReleasesTags

Edit releaseDelete

Latest release

1.0
0c79dee

1.0: 我修改了冲突

yw-forGit released this 39 seconds from now

Assets 2

Source code (zip)

Source code (tar.gz)

1.0版本

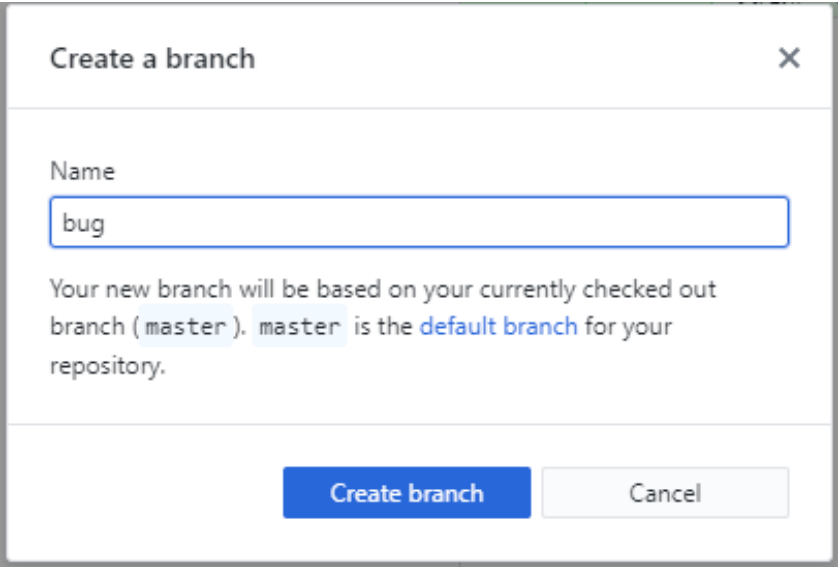
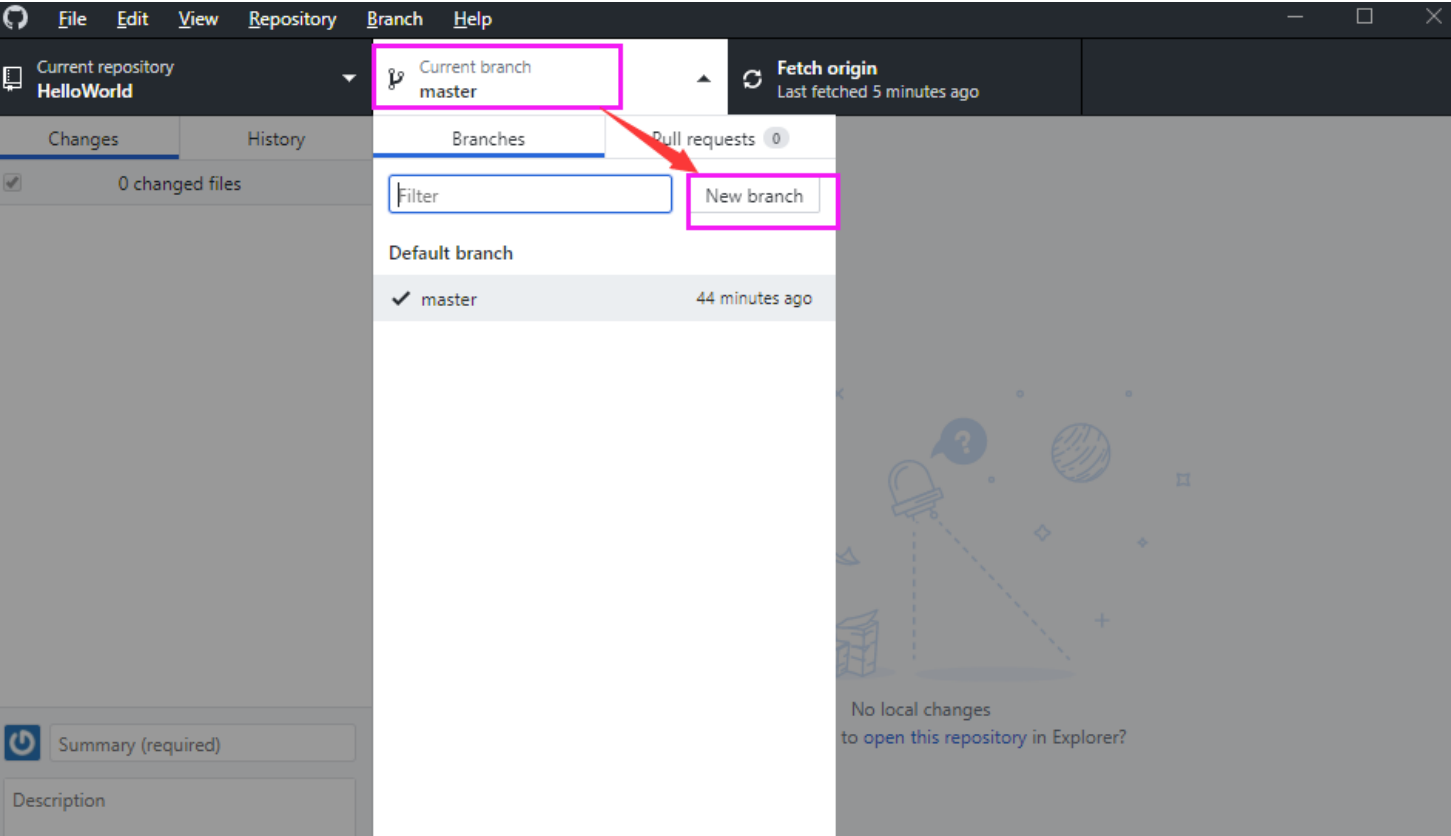
这里就可以看到都更新了什么，还可以自由选择下载哪个版本

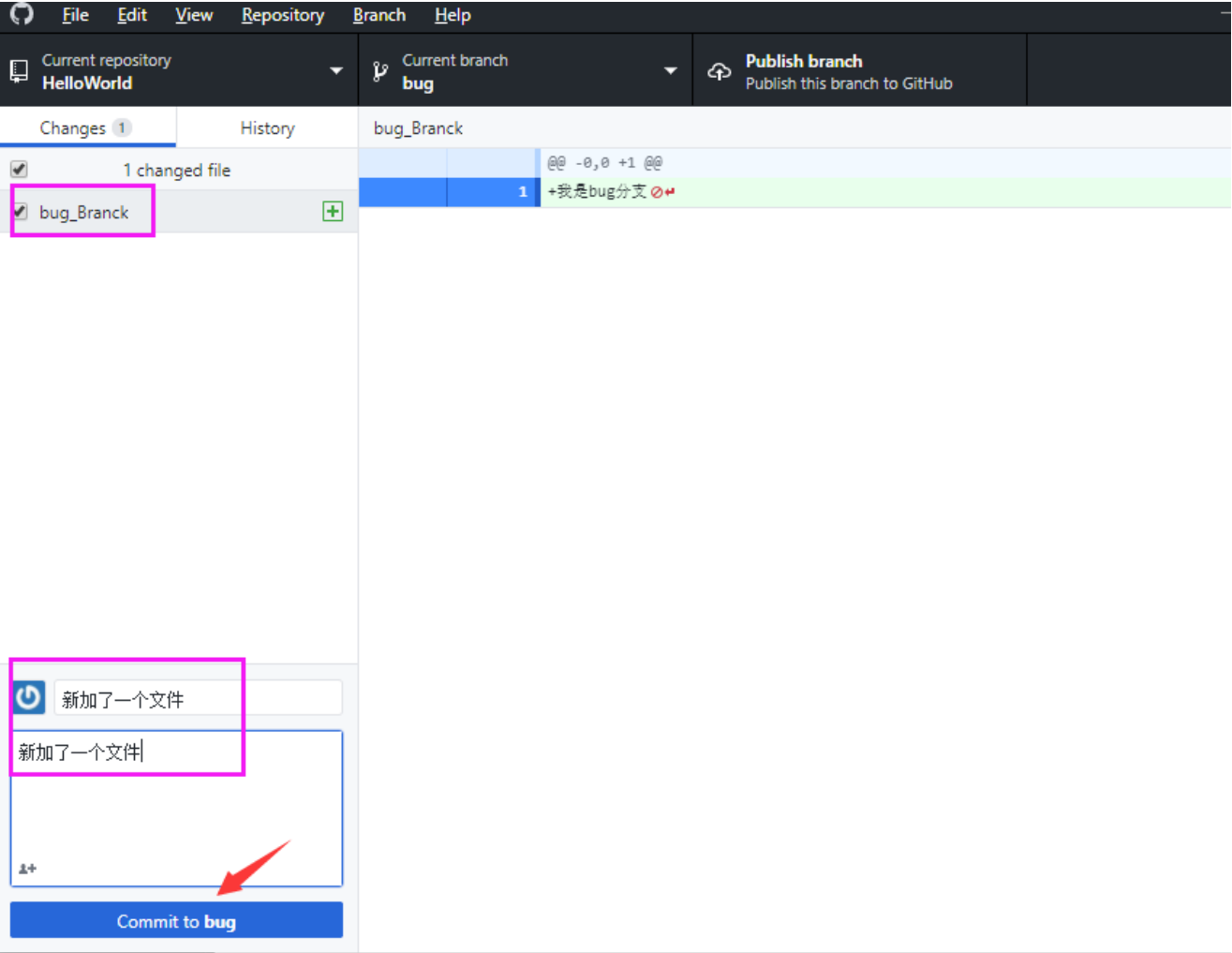
10、分支、分支合并

需要解决紧急的 bug 时，而这时 master 主分支又有新的代码提交，但是还没有测试通过，故可以拿 master 之前的已经验证

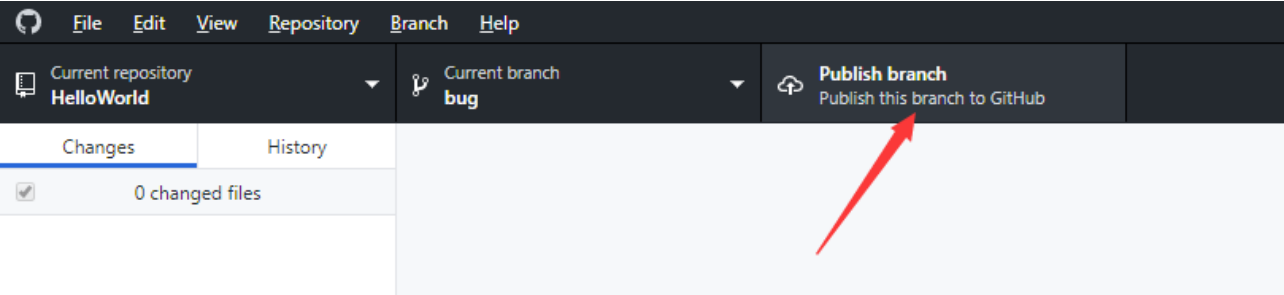
通过的节点代码作为一个新的分支（叫 **bug** 分支吧），然后在此修复紧急 **bug**，就可以越过新提交没有测试过的代码。等待紧急 **bug** 解决了，上线了，没有问题了，就把这个 **bug** 分支合并到 **master** 主分支里面去，这样就不会丢失 **bug** 这个分支所修复的代码。

分支

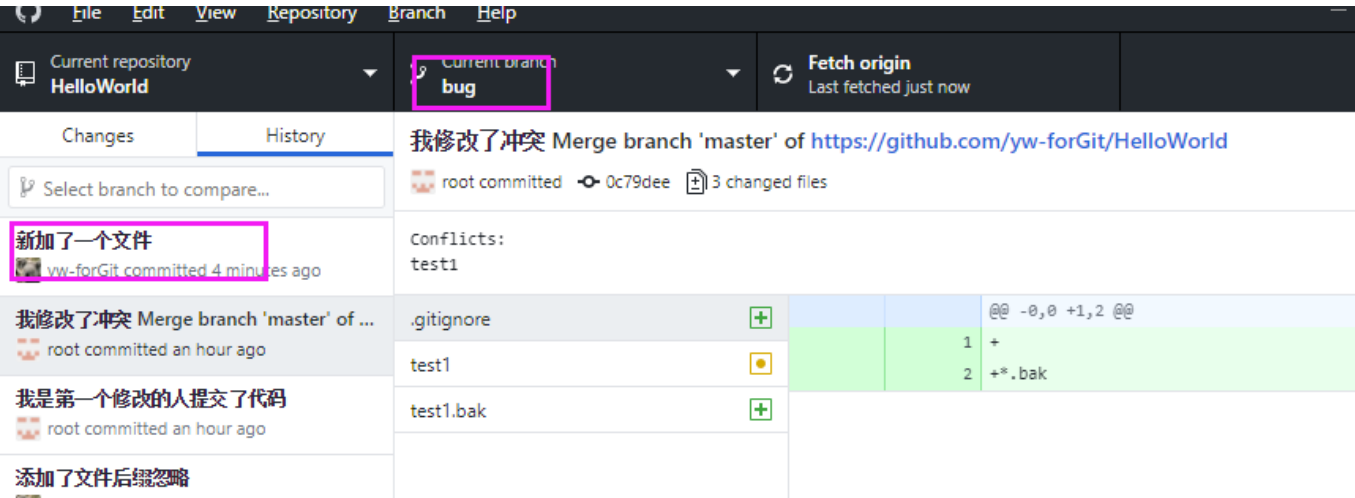




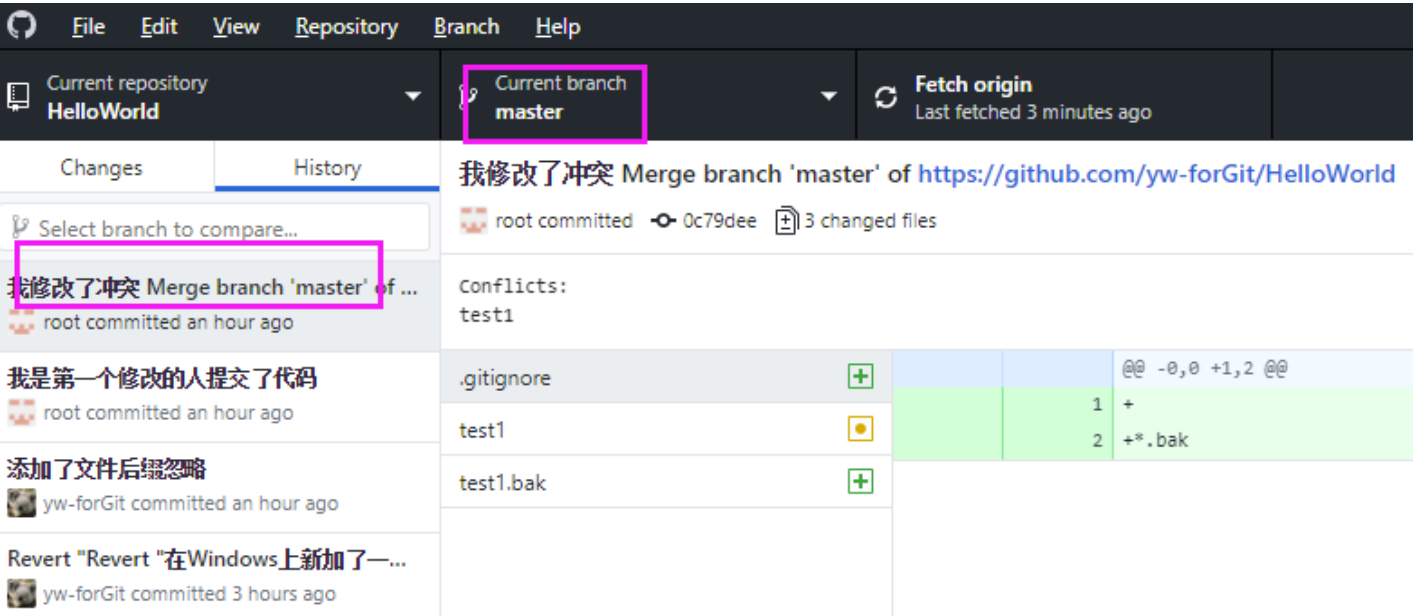
push 到 bug 分支



看到 bug 分支有历史记录

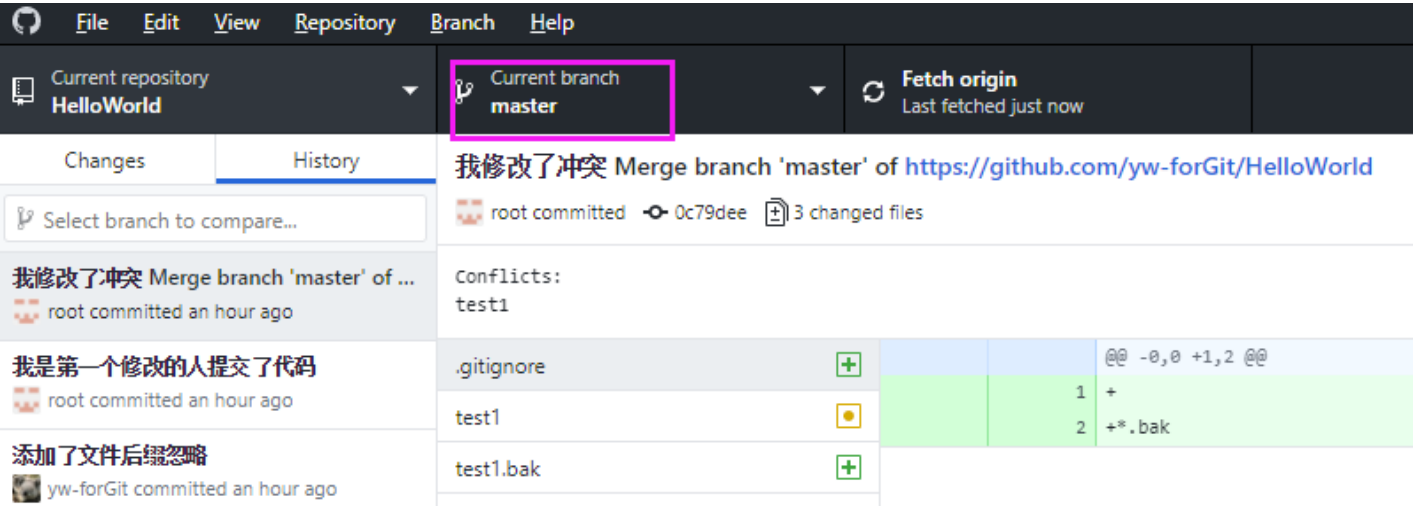


可以看到 master 分支，什么都没有改变

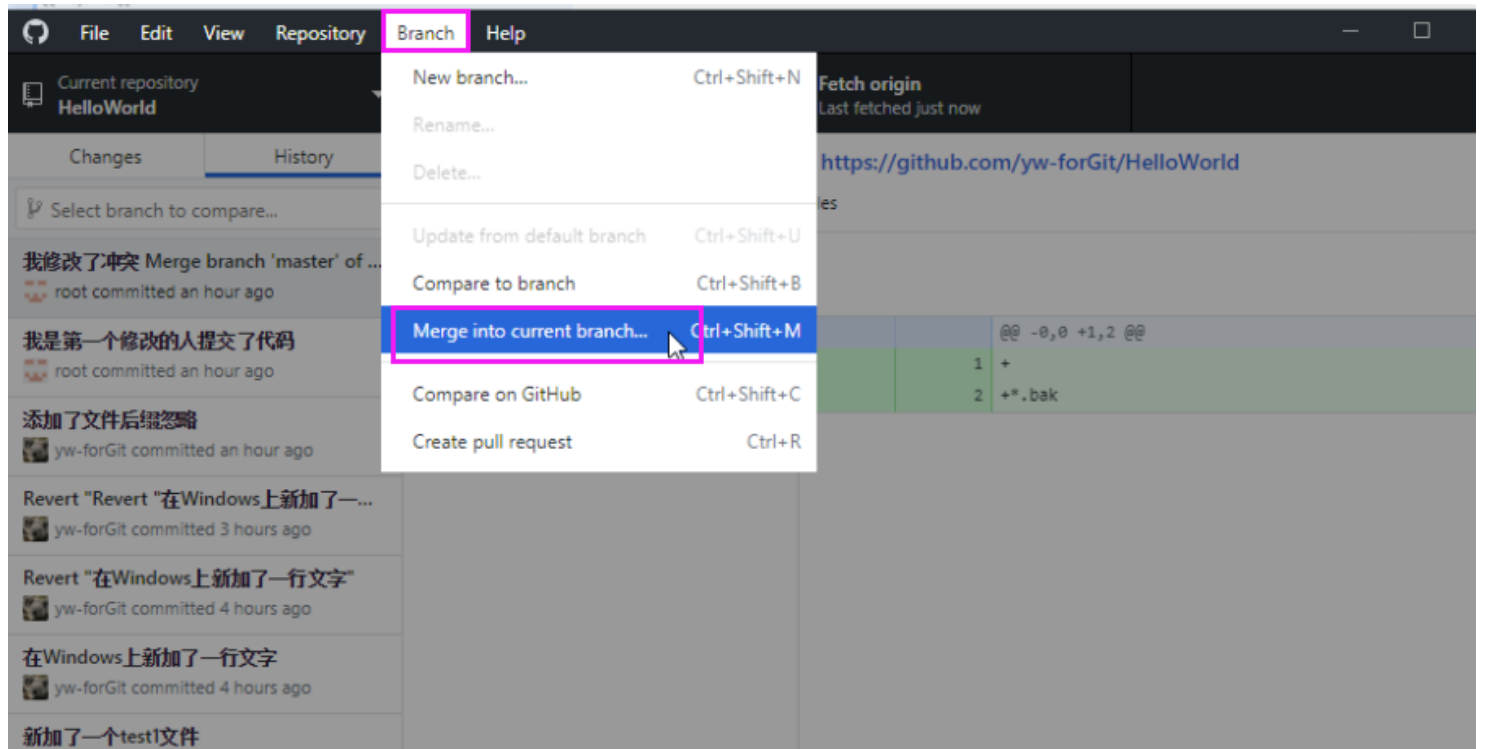


合并分支

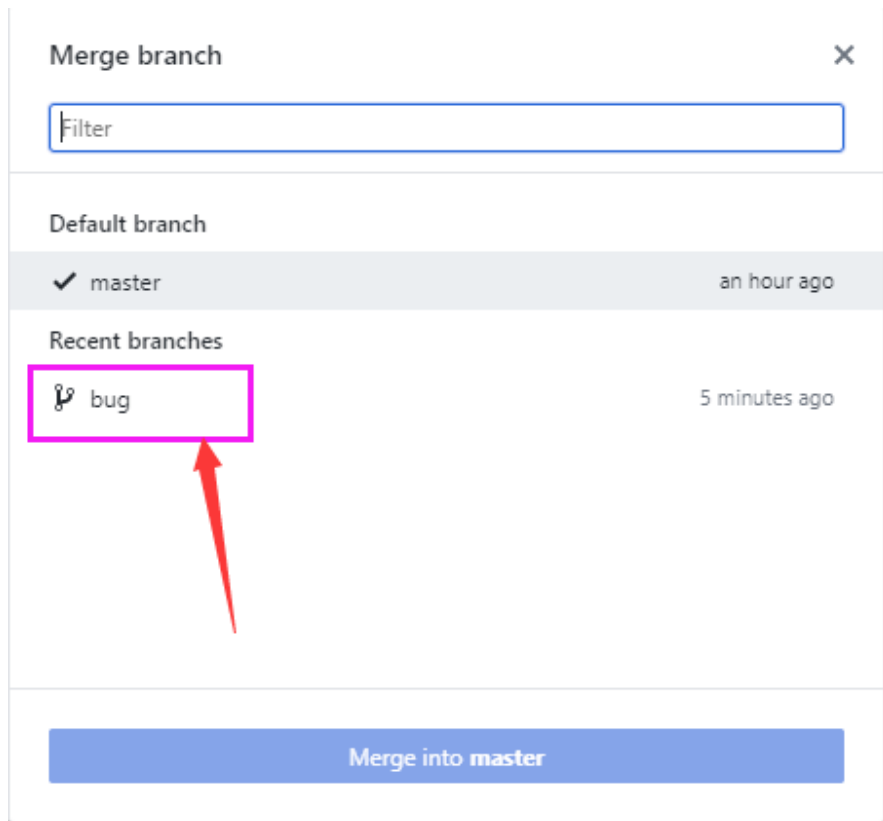
选中 master 分支

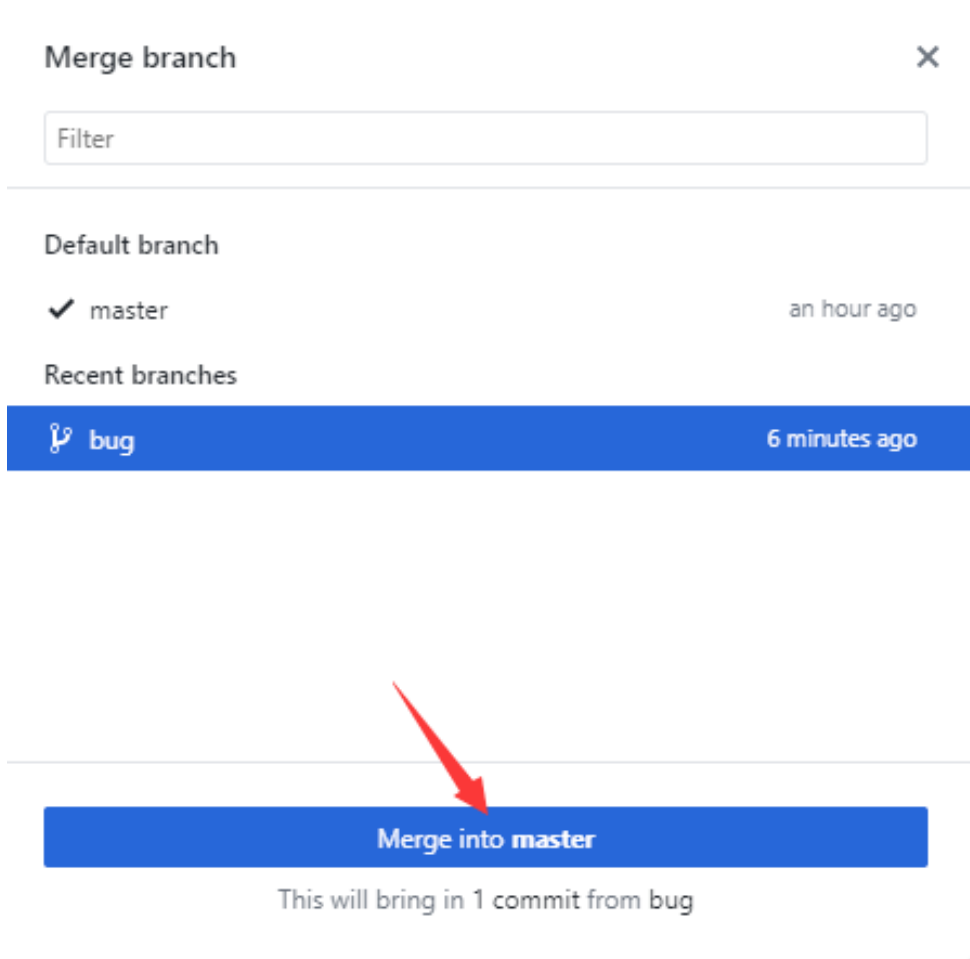


然后准备合并 bug 分支到 master 中

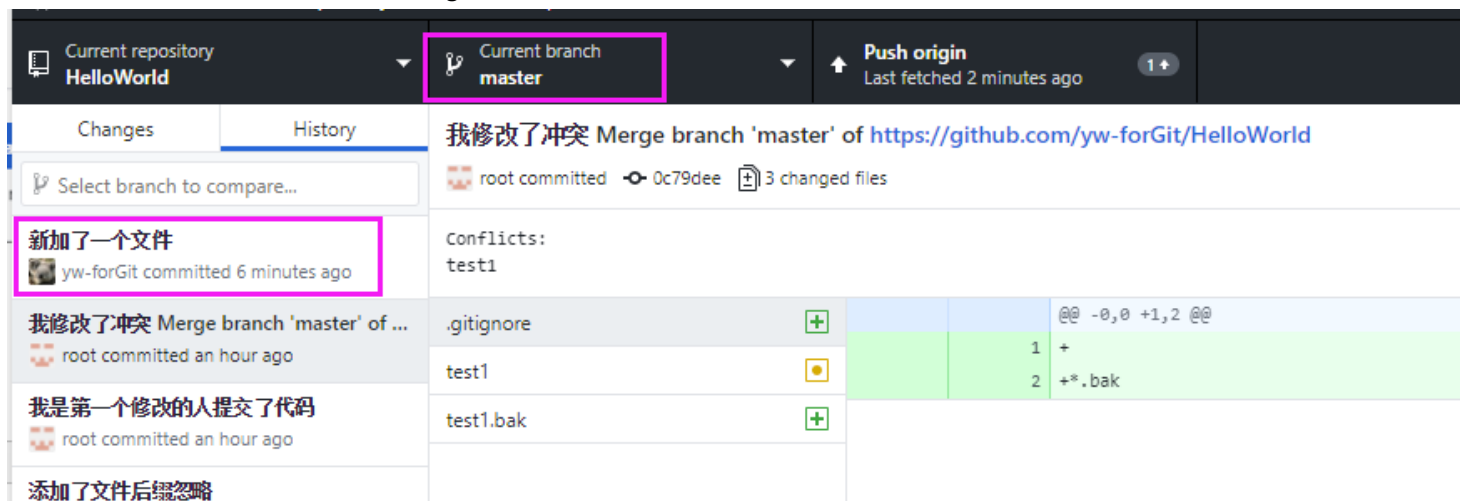


选中被合并的 bug 分支

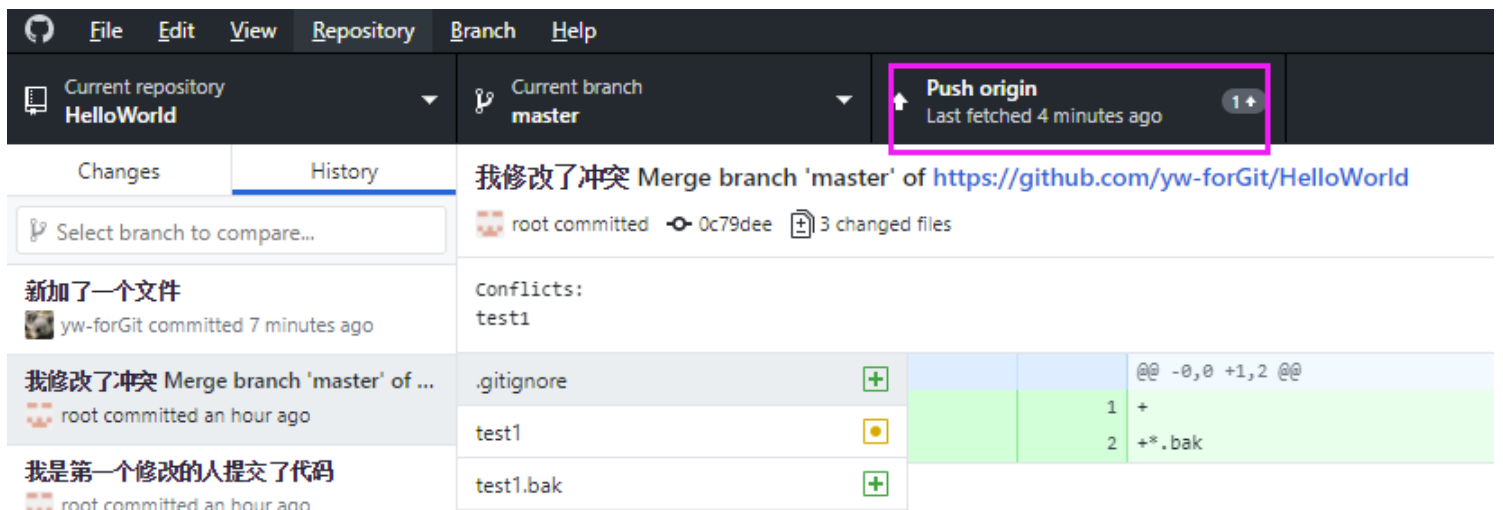




然后就能看到 master 主分支也有 bug 分支的提交了



同步一下



合并成功

CodeIssues 0Pull requests 0Projects 0WikiInsightsSettings

第二次在这里向世界问好啦Edit

Add topics

9 commits2 branches1 release2 contributors

BranchmasterNew pull requestCreate new fileUpload filesFind fileClone or download

yw-forGit 新加了一个文件Latest commit 6704cc1 8 minutes ago

.gitignore	添加了文件后缀忽略	an hour ago
README.md	Initial commit	5 hours ago
bug_Branck	新加了一个文件	8 minutes ago
test1	我修改了冲突	an hour ago
test1.bak	Revert "Revert "在Windows上新加了一行文字"	4 hours ago

README.md

结束日期：2018-09-04