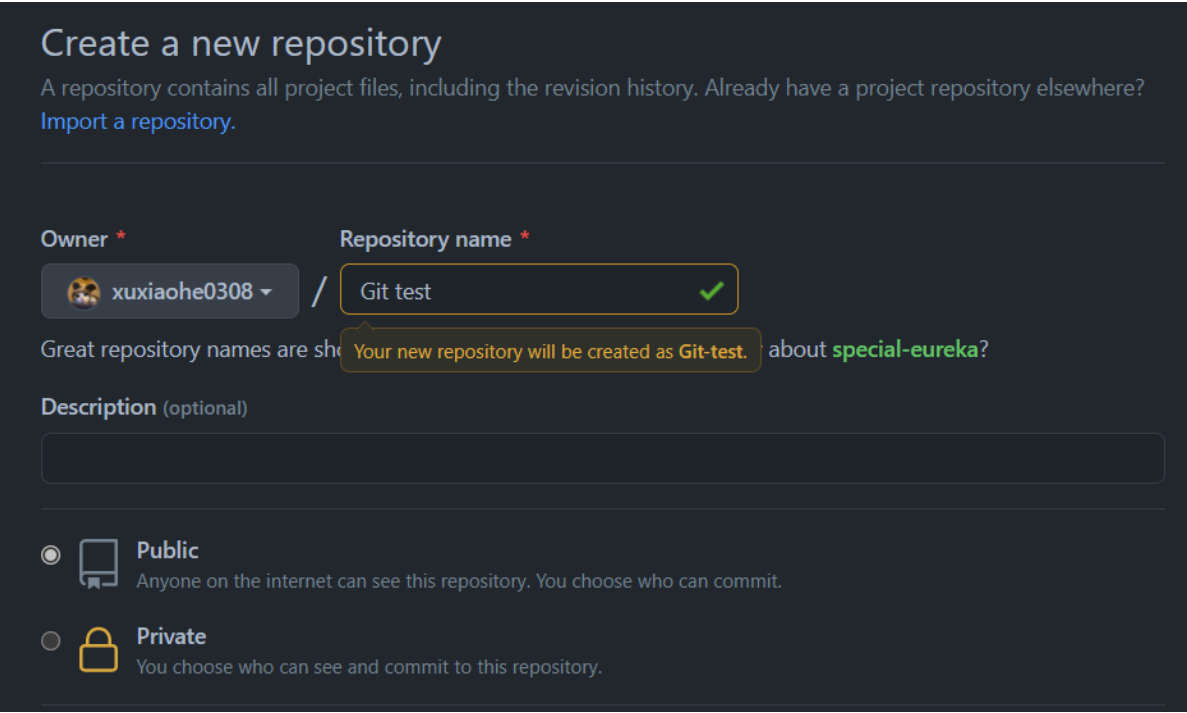


Git使用练习报告

一、创建仓库

首先在Github创建新仓库。



在本地创建文件，列表如下。

```
PS E:\Desktop\文件\Git repository\Git test> ls

    目录: E:\Desktop\文件\Git repository\Git test

Mode                LastWriteTime         Length Name
----                -
d-----          2021/7/16      15:32           figure
-a-----          2021/7/16      15:34             16 .gitignore
-a-----          2021/7/16      12:50      963593 Git的使用与管理.pptx
-a-----          2021/7/16      15:32             0 README.md
-a-----          2021/7/16      15:29             0 record.docx

PS E:\Desktop\文件\Git repository\Git test>
```

其中 .gitignore 用于指定被git忽略的PPT等文件，这些文件不会被git所追踪。文件内容如下。

```
/* .pptx
/* .docx
```

之后在此路径中打开shell，输入命令新建本地git仓库。

```
git init
```

查看目前文件状态。

```
git status
```

```
PS E:\Desktop\文件\Git repository\Git test> git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore
        README.md

nothing added to commit but untracked files present (use "git add" to track)
```

两个文件为红色状态，代表文件未追踪。下一步需要使用 `add` 命令使git追踪这些文件。可以使用：

```
git add .\gitignore .\README.md
```

或

```
git add .
```

(后者用于一次性追踪所有文件。)

`add` 完成后所有文件状态显示为绿色。

```
PS E:\Desktop\文件\Git repository\Git test> git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   .gitignore
        new file:   README.md
```

之后进行提交。

```
git commit
```

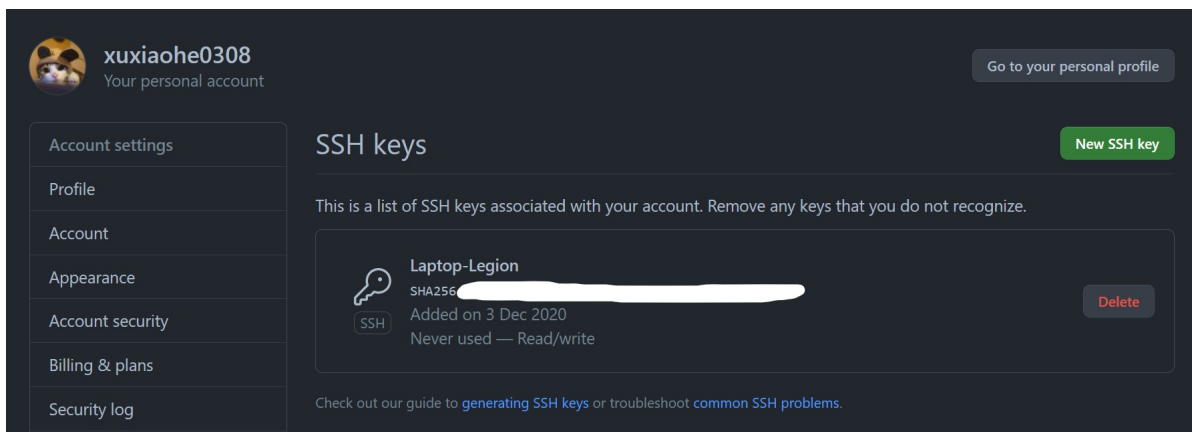
系统会弹出默认文件编辑器，在其中编辑提交信息，然后保存关闭。

```
*E:\Desktop\文件\Git repository\Git test\.git\COMMIT_EDITMSG - Notepad++
文件(F) 编辑(E) 搜索(S) 视图(V) 编码(N) 语言(L) 设置(T) 工具(O) 宏(M) 运行(R) 插件(P) 窗口(W) ?

1 Init commit
2 # Please enter the commit message for your changes. Lines starting
3 # with '#' will be ignored, and an empty message aborts the commit.
4 #
5 # On branch master
6 #
7 # Initial commit
8 #
9 # Changes to be committed:
10 #   new file:   .gitignore
11 #   new file:   README.md
12 #
13
```

```
PS E:\Desktop\文件\Git repository\Git test> git commit
[master (root-commit) 183315e] Init commit
2 files changed, 3 insertions(+)
create mode 100644 .gitignore
create mode 100644 README.md
```

下一步需要将本地的提交推送至github对应的远程仓库。本示例在github中导入了主机的SSH公钥，方便后续免密推送（这里展示添加后的结果）。



下一步在github中建立远程源，并将本地的仓库推送上去。`-u` 命令指定了本地main分支和远程origin分支的配对，后续进行 `push` 和 `pull` 等操作时无需指定分支信息。

```
git remote add origin https://github.com/xuxiaohe0308/Git-test.git
git branch -M main
git push -u origin main
```

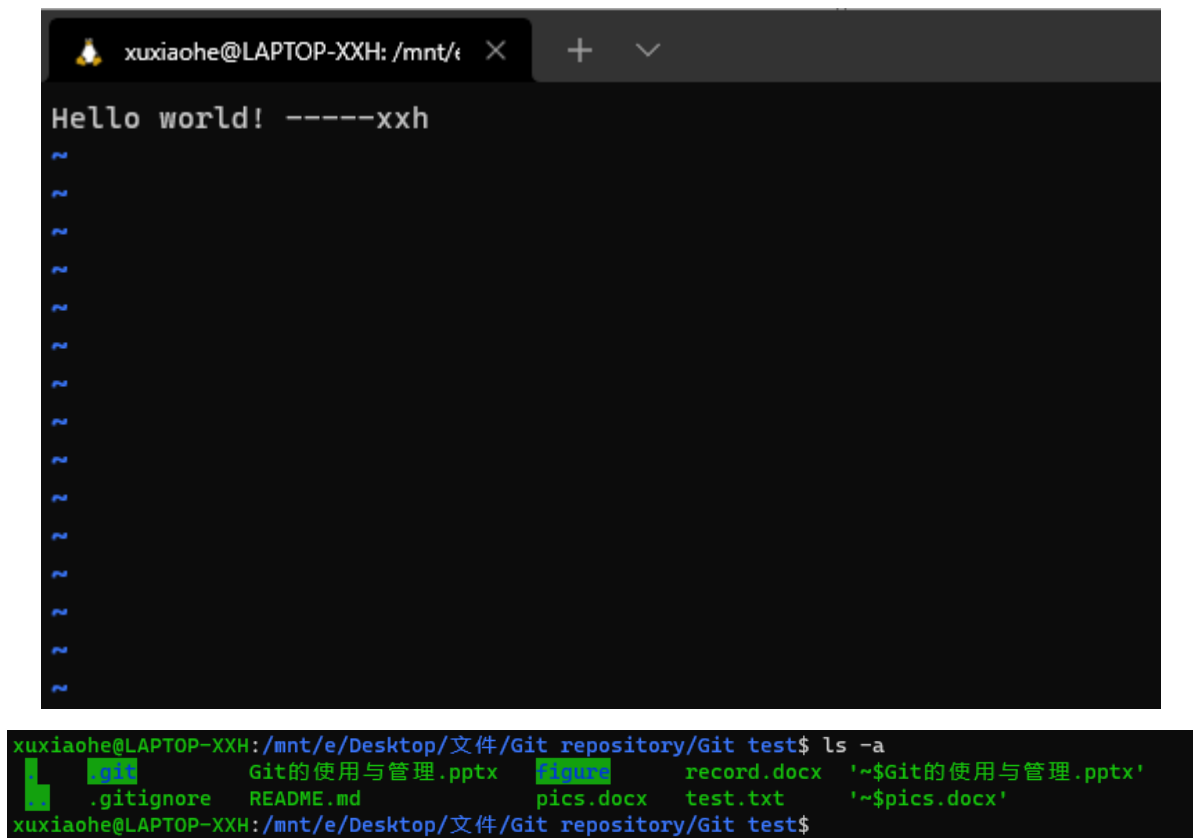
```
PS E:\Desktop\文件\Git repository\Git test> git remote add origin https://github.com/xuxiaohe0308/Git-test.git
PS E:\Desktop\文件\Git repository\Git test> git branch -M main
PS E:\Desktop\文件\Git repository\Git test> git push -u origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 318 bytes | 318.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/xuxiaohe0308/Git-test.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
```

完成后github上同步了本地仓库。



二、上传新文件

本地新建测试文件test.txt。



提交并推送（由于之前已经使用过 `-u` 参数，所以这里 `push` 时无需指定其余参数）。






```
xuxiaohe@LAPTOP-XXH:/mnt/e/Desktop/文件/Git repository/Git test$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   test.txt

xuxiaohe@LAPTOP-XXH:/mnt/e/Desktop/文件/Git repository/Git test$ git commit -m 'update test.txt'
[main 89015f9] update test.txt
 1 file changed, 1 insertion(+)
 create mode 100644 test.txt
```

```
86156@LAPTOP-XXH MINGW64 /e/Desktop/文件/Git repository/Git test (main)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 329 bytes | 329.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/xuxiaohe0308/Git-test.git
 d4a255a..89015f9  main -> main
```

github端新增test.txt文件。

	xuxiaohe0308 update test.txt	89015f9 4 minutes ago	 3 commits
	.gitignore	Init commit	12 minutes ago
	README.md	Init commit	9 days ago
	test.txt	update test.txt	4 minutes ago

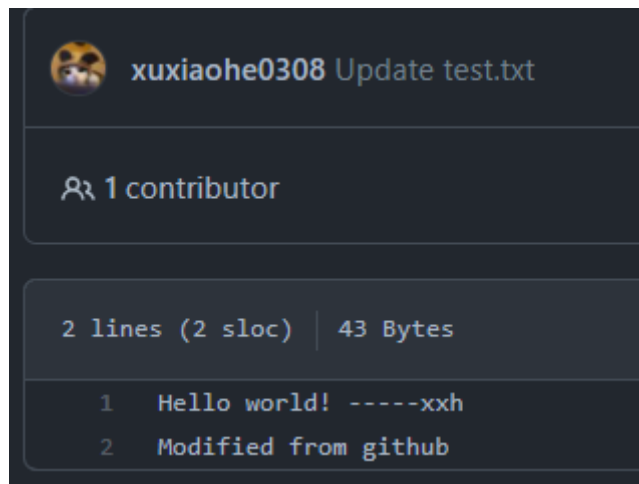
三、解决冲突

本示例模拟了一种实际代码管理场景中会出现的冲突。

- 用户在本地对本地仓库的文件test.txt进行了修改，添加了一行文字 Modified from local；

```
test.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
Hello world! -----xxh
Modified from local
```

- 与此同时其他用户在他们的本地进行了修改并推送到了远程仓库，修改了远程仓库的内容（添加了一行文字 Modified from github），本示例直接在github上进行修改来模拟此过程。



上述冲突发生的时候，用户使用 `git push` 时会提示被拒绝，并建议首先同步远程仓库的内容。

```
86156@LAPTOP-XXH MINGW64 /e/Desktop/文件/Git repository/Git test (main)
$ git push
To https://github.com/xuxiaohe0308/Git-test.git
! [rejected]        main -> main (fetch first)
error: failed to push some refs to 'https://github.com/xuxiaohe0308/Git-test.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

`git pull`

```
86156@LAPTOP-XXH MINGW64 /e/Desktop/文件/Git repository/Git test (main)
$ git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 672 bytes | 84.00 KiB/s, done.
From https://github.com/xuxiaohe0308/Git-test
 89015f9..79d591d  main      -> origin/main
Auto-merging test.txt
CONFLICT (content): Merge conflict in test.txt
Automatic merge failed; fix conflicts and then commit the result.

86156@LAPTOP-XXH MINGW64 /e/Desktop/文件/Git repository/Git test (main|MERGING)
$ git status
On branch main
Your branch and 'origin/main' have diverged,
and have 1 and 1 different commits each, respectively.
(use "git pull" to merge the remote branch into yours)

You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add <file>..." to mark resolution)
    both modified:   test.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

提示远程与本地分支有区别，需要合并。使用文本编辑器打开本地文件查看冲突。

```
test.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
Hello world! -----xxh
<<<<<<< HEAD
Modified from local
=====
Modified from github
>>>>>>> 79d591d2947bbb9fe7f287f6d429b43fa877a18a
```

冲突解释:

- <<<<<<<到=====是在当前分支合并之前的文件内容;
- =====到>>>>>>>是在其它分支下修改的内容。


根据需求人工解决冲突。

```
test.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
Hello world! -----xxh
Modified from local & github
```

```
86156@LAPTOP-XXH MINGW64 /e/Desktop/文件/Git repository/Git test (main|MERGING)
$ git add .

86156@LAPTOP-XXH MINGW64 /e/Desktop/文件/Git repository/Git test (main|MERGING)
$ git commit -m 'solving conflict'
[main 345425b] solving conflict

86156@LAPTOP-XXH MINGW64 /e/Desktop/文件/Git repository/Git test (main)
$ git push
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (6/6), 627 bytes | 627.00 KiB/s, done.
Total 6 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To https://github.com/xuxiaohe0308/Git-test.git
79d591d..345425b main -> main
```

 **xuxiaohe0308** solving conflict

1 contributor

3 lines (2 sloc) | 52 Bytes

```
1 Hello world! -----xxh
2 Modified from local & github
3
```

可以看到此时的提交、推送等操作可以执行，并且成功地更新了github上的内容。

四、版本回退

git提供两种版本回退的方式，分别为：

- `git reset`，此命令用于移动当前版本指针至之前的某一个版本，**会删除回退版本之后所有版本的记录**；
- `git revert`，此命令用于还原之前某一版本的修改内容，并生成一个新的提交，**不会删除之前版本的记录，推荐使用此方式**。

假设用户不满意所有对test.txt文件的修改，希望还原到第一次提交test.txt的版本，但同时不能删除这些修改记录，那么他就需要使用版本回退功能。

首先查看历史提交的版本号。

```
git log
```

```
86156@LAPTOP-XXH MINGW64 /e/Desktop/文件/Git repository/Git test (main)
$ git log
commit 345425bf47ad5e29e2a37a75ff318a27dd50b9bd (HEAD -> main, origin/main)
Merge: b9a112a 79d591d
Author: xuxiaohe0308 <58452249+xuxiaohe0308@users.noreply.github.com>
Date: Sun Jul 25 20:53:54 2021 +0800

    solving conflict
    是在当前分支合并之前的文件内容。
commit b9a112a1b3dbee00d2838ef25e6672f9e388f709
Author: xuxiaohe0308 <58452249+xuxiaohe0308@users.noreply.github.com>
Date: Sun Jul 25 20:46:26 2021 +0800

    update test from local
    是在合并前，修改到本地（b9a112a）
! -- Update test.txt
commit 79d591d2947bbb9fe7f287f6d429b43fa877a18a
Author: xuxiaohe0308 <58452249+xuxiaohe0308@users.noreply.github.com>
Date: Sun Jul 25 20:44:53 2021 +0800

    update test.txt
    决冲突
! -- Update test.txt
commit 89015f950956027bdd02ea26baaa841768187b66
Author: xuxiaohe0308 <xuxiaohe@buaa.edu.cn>
Date: Sun Jul 25 20:37:53 2021 +0800

    update test.txt
    在本地? github
commit d4a255ab7cbad3121a3417a41fcde19121002d5a
Author: xuxiaohe0308 <58452249+xuxiaohe0308@users.noreply.github.com>
Date: Sun Jul 25 20:30:24 2021 +0800

    Init commit
    MINGW64 /e/Desktop/文件/Git repository/Git test (main|MERGING)
commit 183315e9b5ffedebc398c1ecc59d63056d476d01 (main|MERGING)
Author: xuxiaohe0308 <58452249+xuxiaohe0308@users.noreply.github.com>
Date: Fri Jul 16 15:43:45 2021 +0800

    solving conflict
    MINGW64 /e/Desktop/文件/Git repository/Git test (main)
```

查看提交信息得知，需要回退到20:44:53 2021 +0800时刻提交的updates test.txt处。

```
git revert <版本号>
```



```

86156@LAPTOP-XXH MINGW64 /e/Desktop/文件/Git repository/Git test (main)
$ git revert 79d591d2947bbb9fe7f287f6d429b43fa877a18a
Auto-merging test.txt
CONFLICT (content): Merge conflict in test.txt
error: could not revert 79d591d... Update test.txt
hint: after resolving the conflicts, mark the corrected paths
hint: with 'git add <paths>' or 'git rm <paths>'
hint: and commit the result with 'git commit'

86156@LAPTOP-XXH MINGW64 /e/Desktop/文件/Git repository/Git test (main|REVERTING)
$ git status
On branch main
Your branch is up to date with 'origin/main'

You are currently reverting commit 79d591d.
  (fix conflicts and run "git revert --continue")
  (use "git revert --skip" to skip this patch)
  (use "git revert --abort" to cancel the revert operation)

Unmerged paths:
  (use "git restore --staged <file>..." to unstage)
  (use "git add <file>..." to mark resolution)
    both modified:   test.txt

no changes added to commit (use "git add" and/or "git commit -a")

```

test.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

Hello world! -----xxh

<<<<<< HEAD

Modified from local & github

=====

>>>>>> parent of 79d591d (Update test.txt)

可以发现，在回退时出现了冲突，这是因为：`revert` 命令只会还原一次提交所做出的修改，而默认保留其余提交的修改。在本案例中，由于本次revert期望删除 Modified from local & github 一行，但后续的两次提交（update test from local和solving conflict）其实都是在此行做出了更改，如果遵从此次revert的意愿就会导致后两次提交的失效。故此场景下需要手动解决冲突。

解决冲突（删掉此行）后add，commit，push，并在github上验证结果。

test.txt - 记事本


文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)


Hello world! -----xxh

```
86156@LAPTOP-XXH MINGW64 /e/Desktop/文件/Git repository/Git test (main|REVERTING)
$ git add .

86156@LAPTOP-XXH MINGW64 /e/Desktop/文件/Git repository/Git test (main|REVERTING)
$ git commit -m 'revert'
[main f277845] revert
1 file changed, 1 insertion(+), 3 deletions(-)

86156@LAPTOP-XXH MINGW64 /e/Desktop/文件/Git repository/Git test (main)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 282 bytes | 282.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/xuxiaohe0308/Git-test.git
345425b..f277845 main -> main
```

 **xuxiaohe0308** revert

 1 contributor

1 lines (1 sloc) | 21 Bytes

1 Hello world! -----xxh

此时已经完美回退到了需要的版本。