Git 使用文档

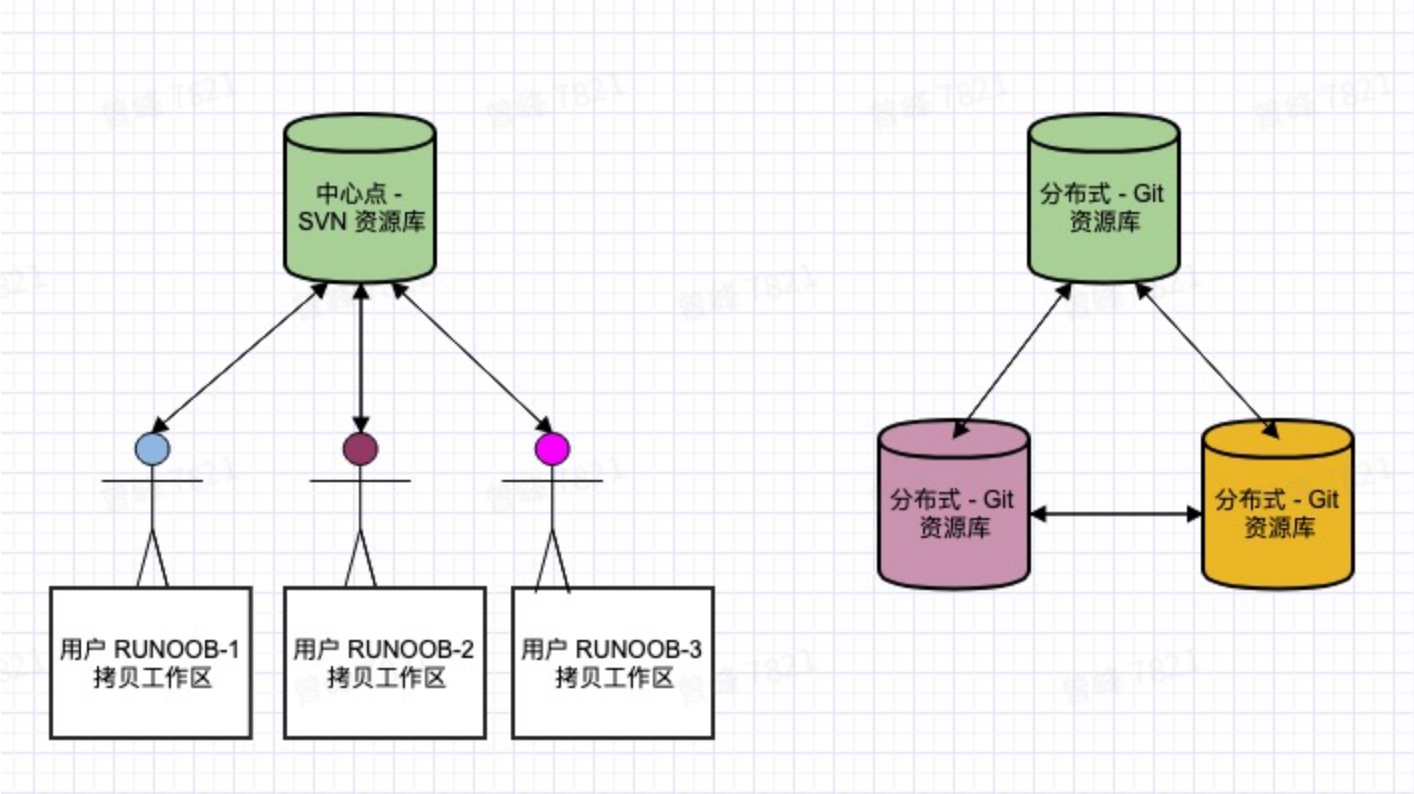
一：Git介绍

https://www.runoob.com/git/git-tutorial.html

1.Git 与 SVN 区别

Git 不仅仅是个版本控制系统，它也是个内容管理系统(CMS)，工作管理系统等。
如果你是一个具有使用 SVN 背景的人，你需要做一定的思想转换，来适应 Git 提供的一些概念和特征。
Git 与 SVN 区别点：

- **1、Git 是分布式的，SVN 不是**：这是 Git 和其它非分布式的版本控制系统，例如 SVN，CVS 等，最核心的区别。
- **2、Git 把内容按元数据方式存储，而 SVN 是按文件：**所有的资源控制系统都是把文件的元信息隐藏在一个类似 .svn、.cvs 等的文件夹里。
- **3、Git 分支和 SVN 的分支不同：**分支在 SVN 中一点都不特别，其实它就是版本库中的另外一个目录。
- **4、Git 没有一个全局的版本号，而 SVN 有：**目前为止这是跟 SVN 相比 Git 缺少的最大的一个特征。
- **5、Git 的内容完整性要优于 SVN：**Git 的内容存储使用的是 SHA-1 哈希算法。这能确保代码内容的完整性，确保在遇到磁盘故障和网络问题时降低对版本库的破坏。



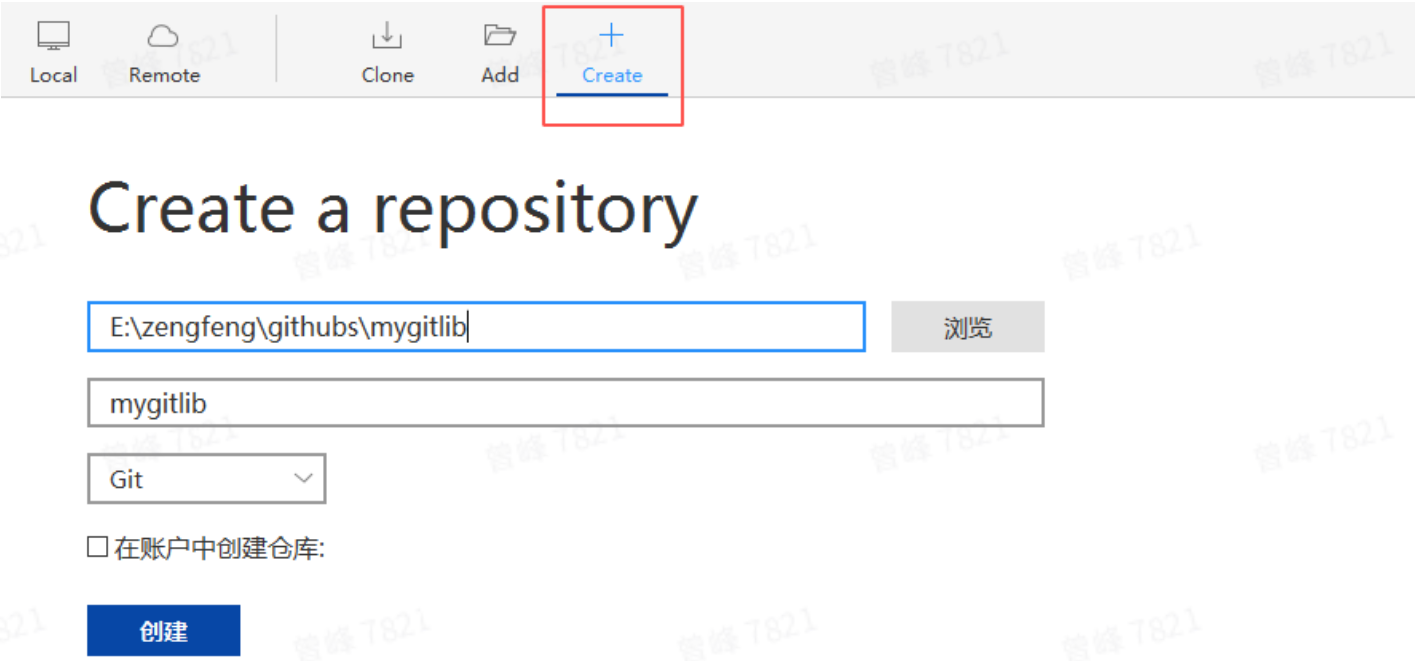
- Git是以仓库为整体clone到本地。不能像SVN以某个目录单独checkout到本地。
- Git是以分支方式提交。SVN是以文件提交。
- Git提交完只是提交到本地仓库，还需要推送到服务器仓库。SVN提交就直接提交到了服务器仓库。

2.如何创建仓库。

我们项目用第三种方式，公司内网gitlab。

方式一：本地创建仓库

安装过git，就可以在本地创建仓库。这种方式基本不怎么用。



方式二：git服务平台

常用的git服务平台有：

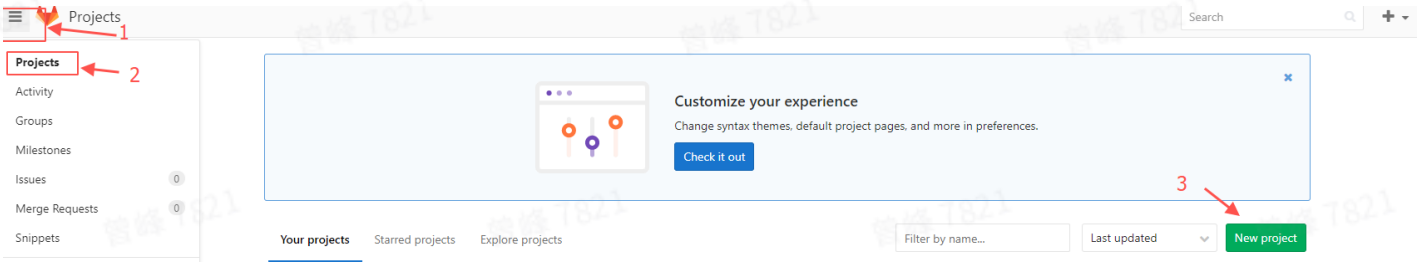
国际 github：https://github.com/

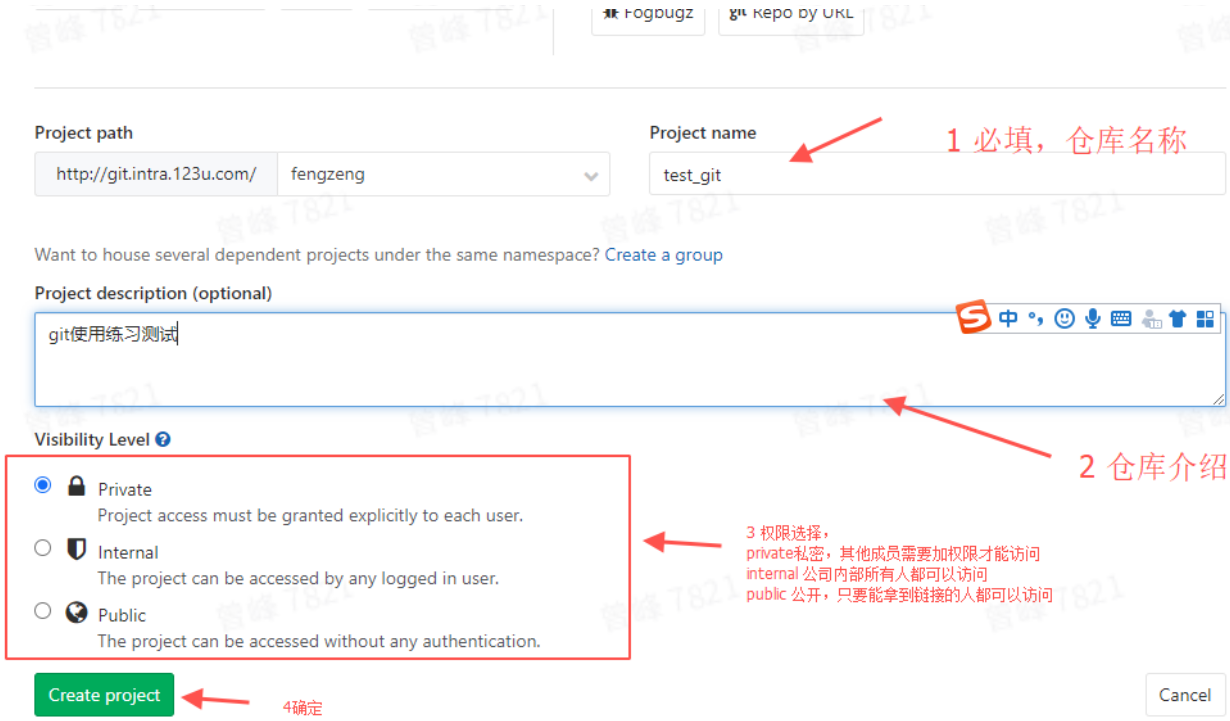
国内 码云: https://gitee.com/

方式三：gitlab

公司一般使用gitlab在公司自己内网搭建服务

http://git.intra.123u.com/

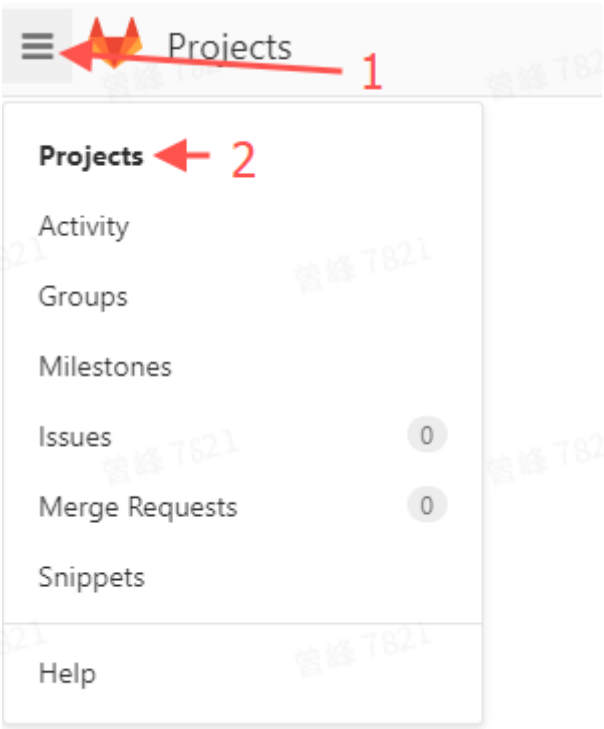




3. 查看自己拥有权限的仓库

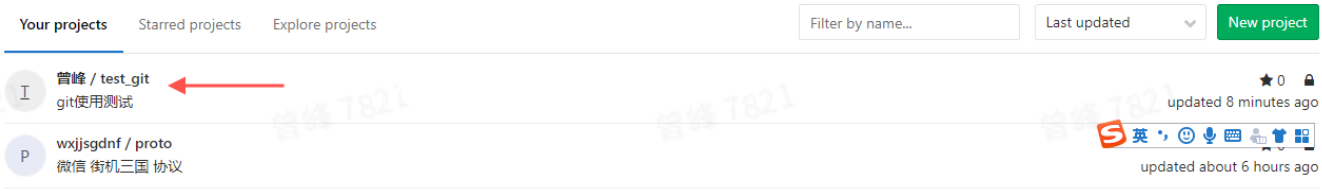
http://git.intra.123u.com/dashboard/projects

登录公司gitlab后。

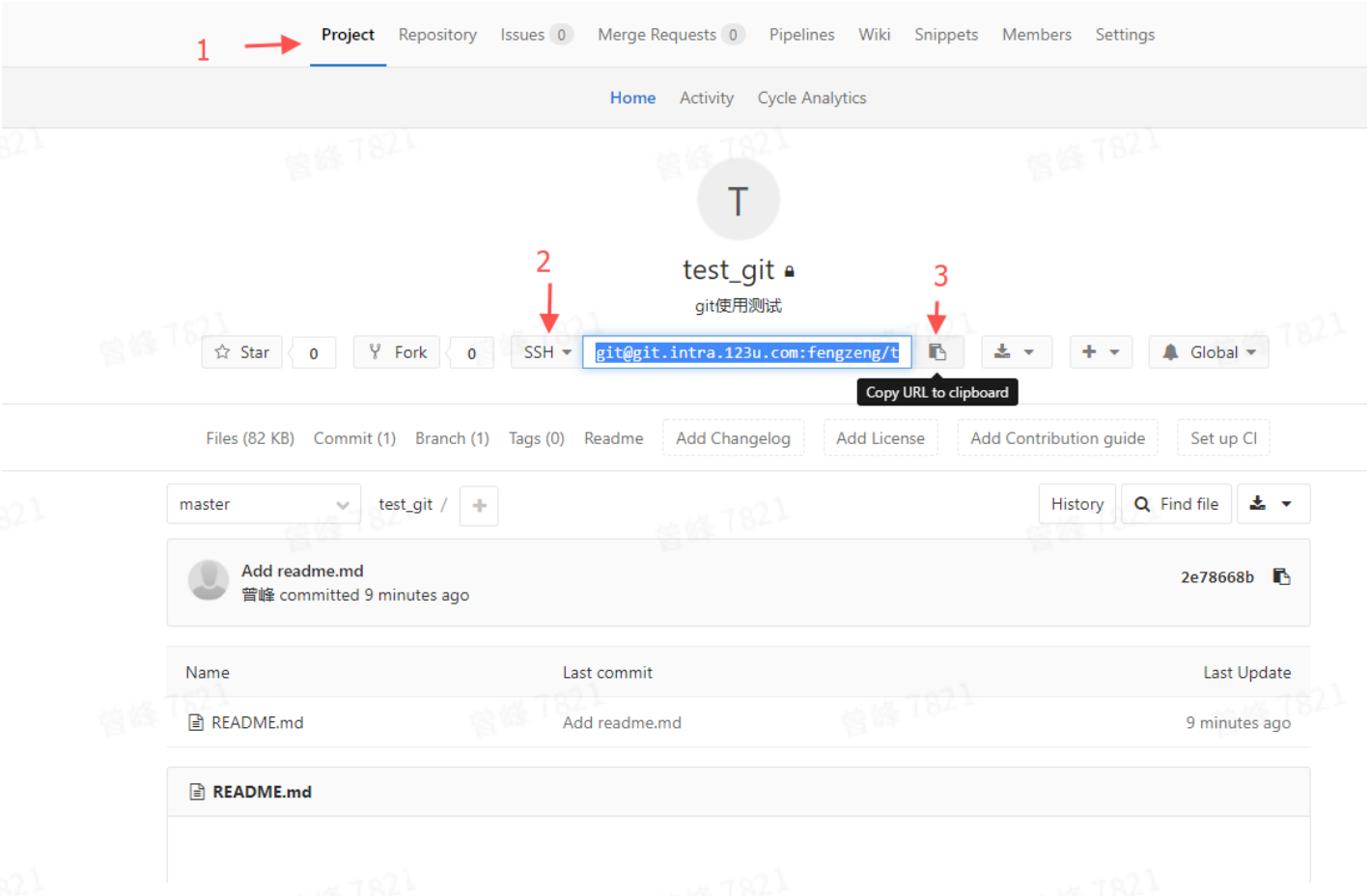


4. 查看仓库地址

点击进入仓库



按步骤，点击复制路径按键就可以复制路径



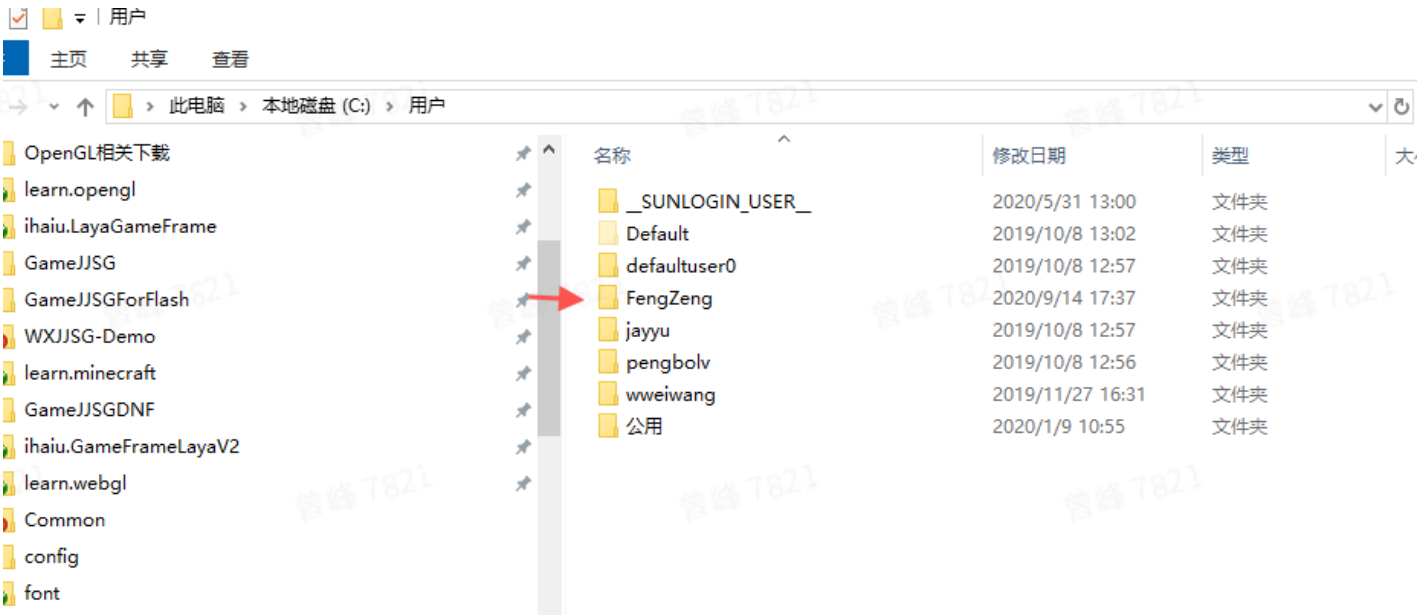
这里介绍一下2种路径的区别:

- HTTP: 在本地克隆仓库的时候需要输入账号和密码。
- SSH: 需要配置公钥，以后本地做git操作就不用每次输入账号密码了。

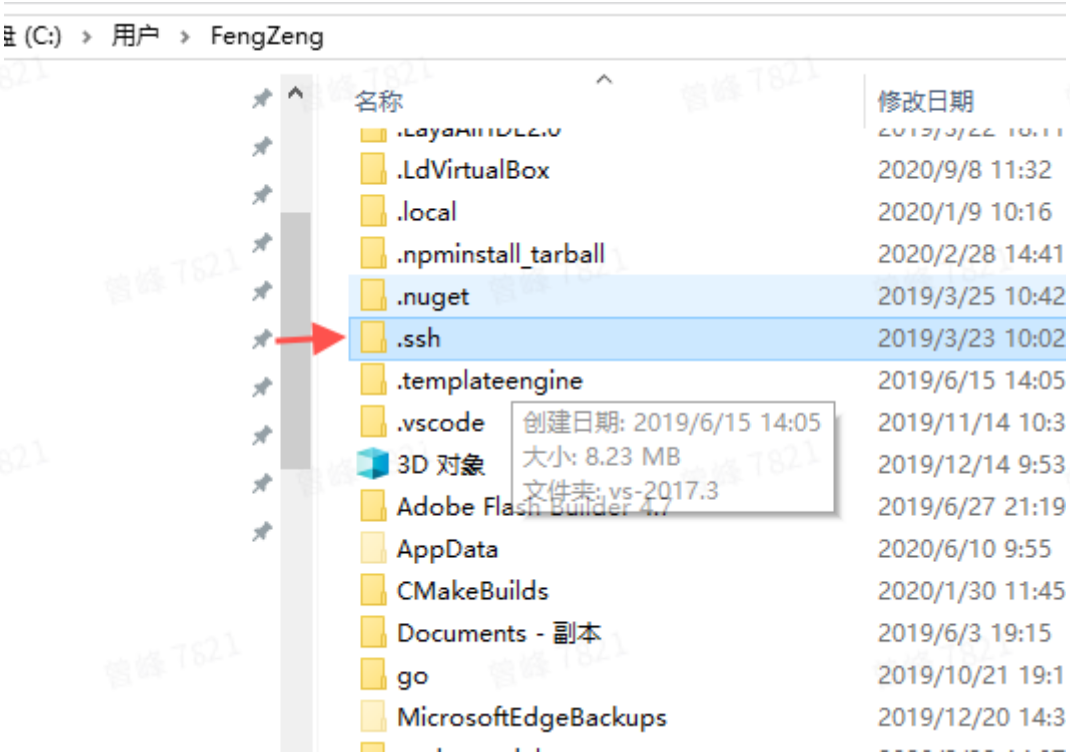
5. 公钥配置

检测本地是否有sshkey

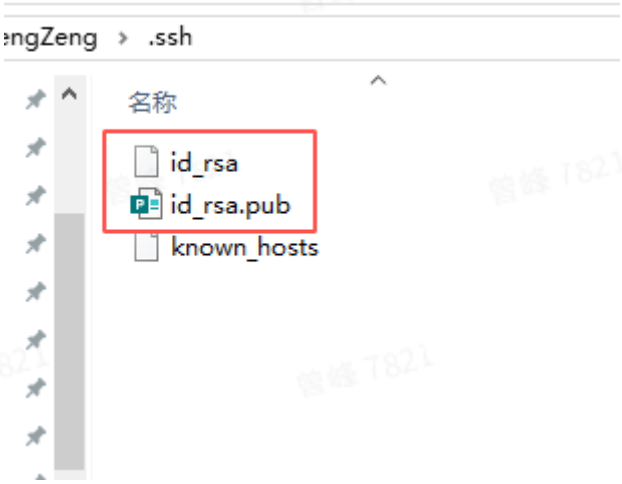
进入用户文档目录下，C:\Users， 再进入到自己用户名目录。



检测是否存在.ssh文件夹。



如果有这2个文件说明已经创建过秘钥id_rsa和公钥id_rsa.pub



创建sshkey

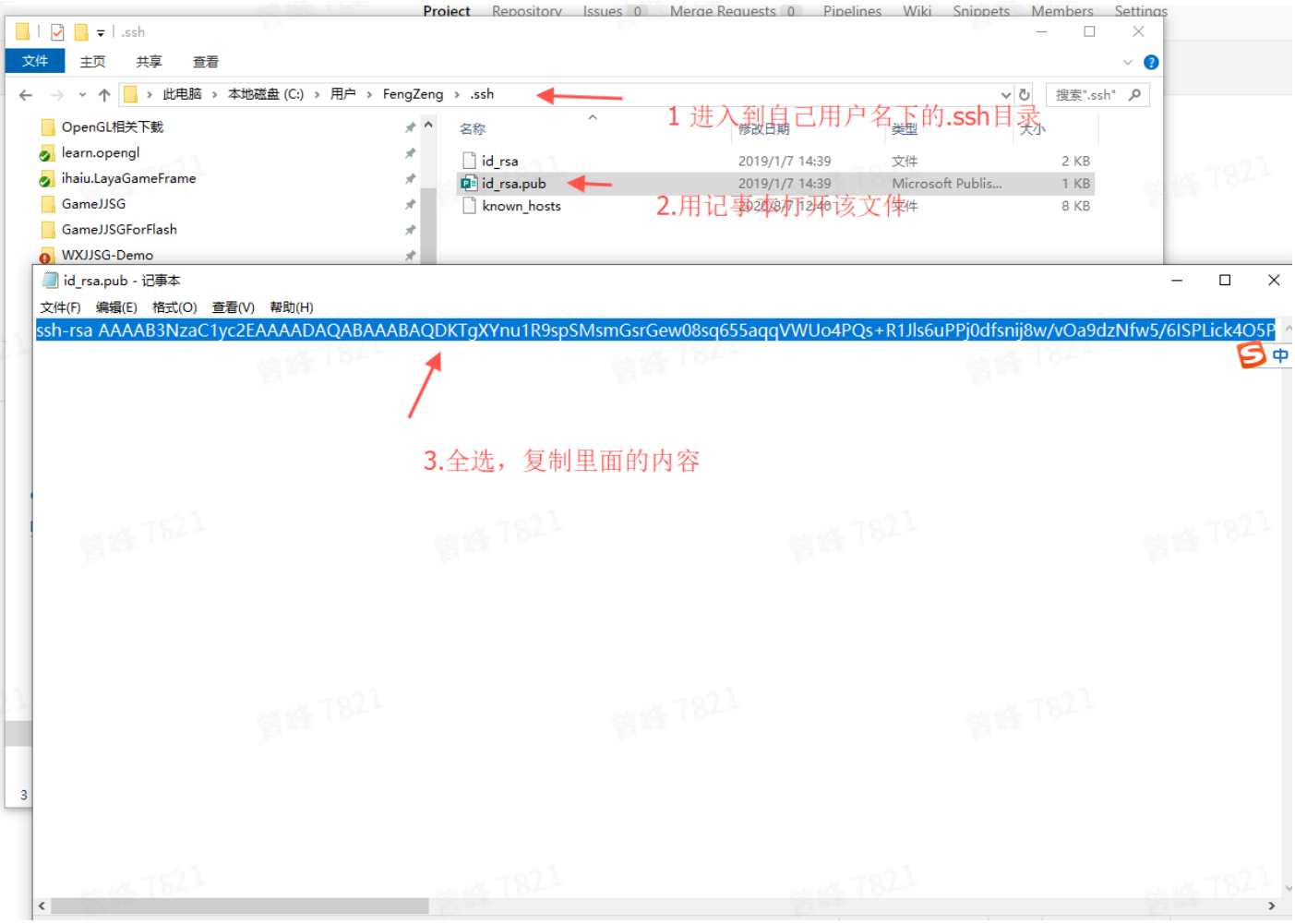
(1) 打开git bash

(2) 输入以下命令

```
cd ~
mkdir .ssh
cd .ssh
## 输入以下命令后 连续点几次回车键
ssh-keygen -t rsa -C"这里写自己的用户名"
```

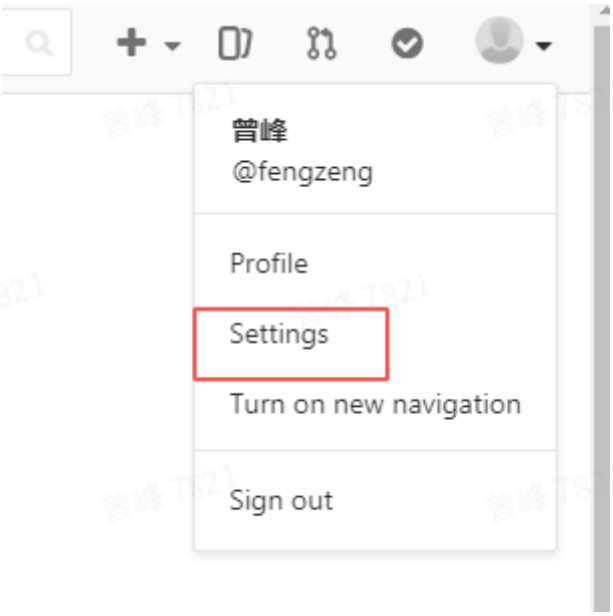
在gitlab上配置公钥

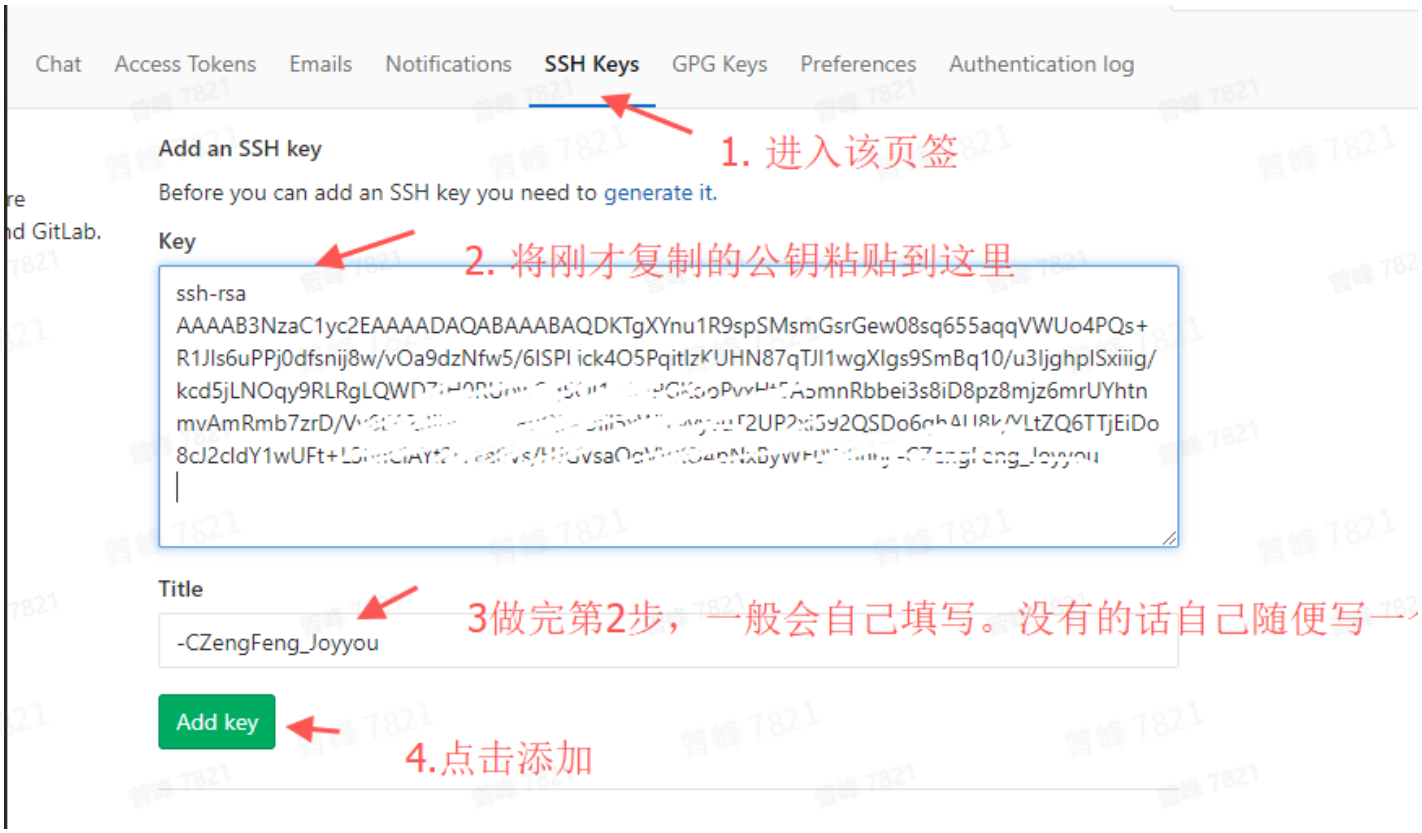
用文本编辑器打开id_rsa.pub，拷贝里面的内容



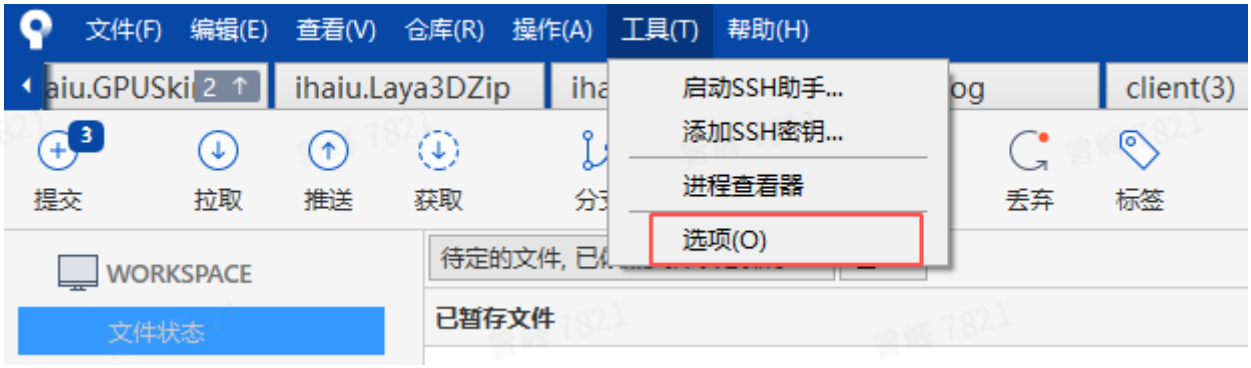
进入gitlab添加ssh key

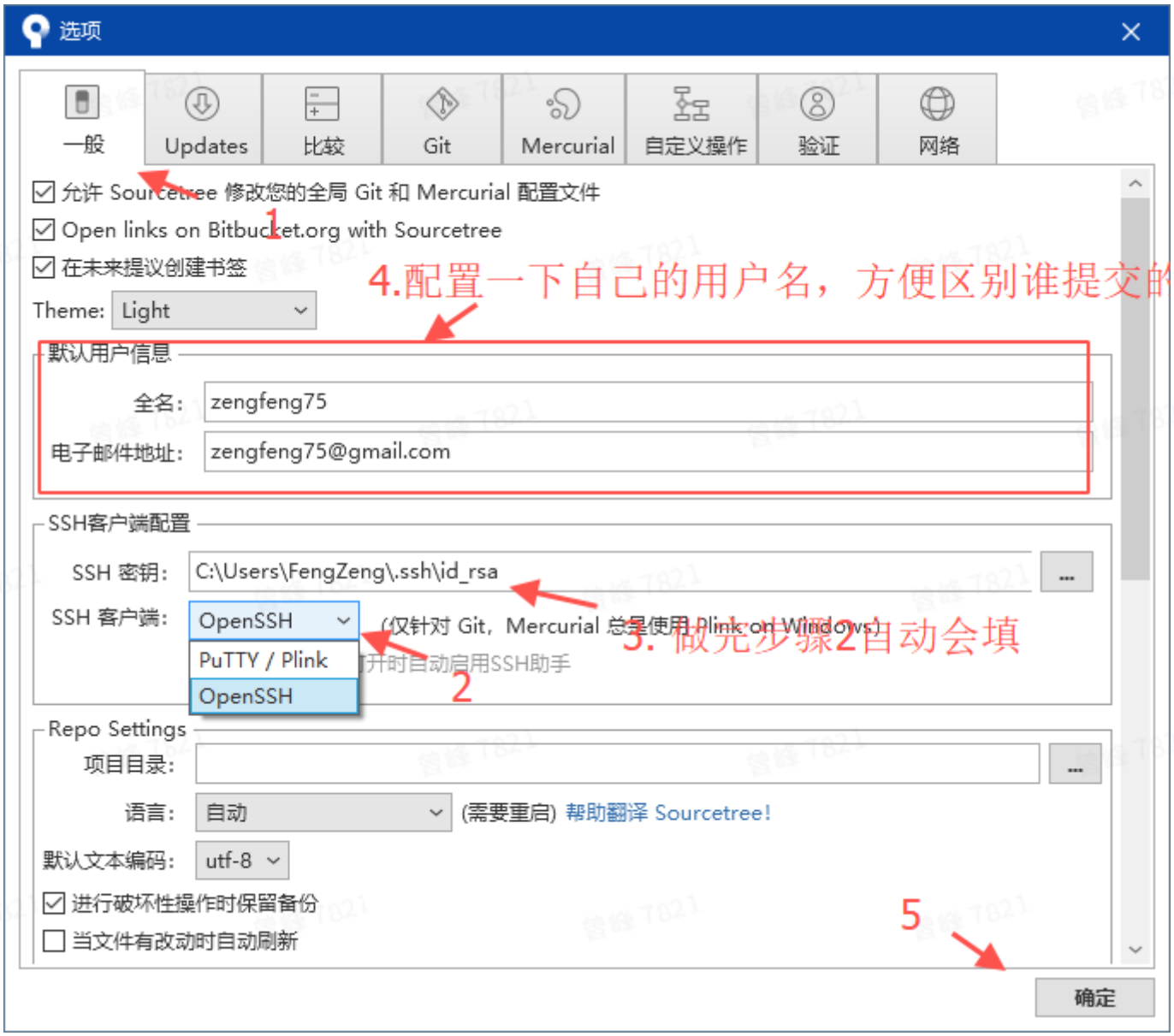
http://git.intra.123u.com/profile/keys



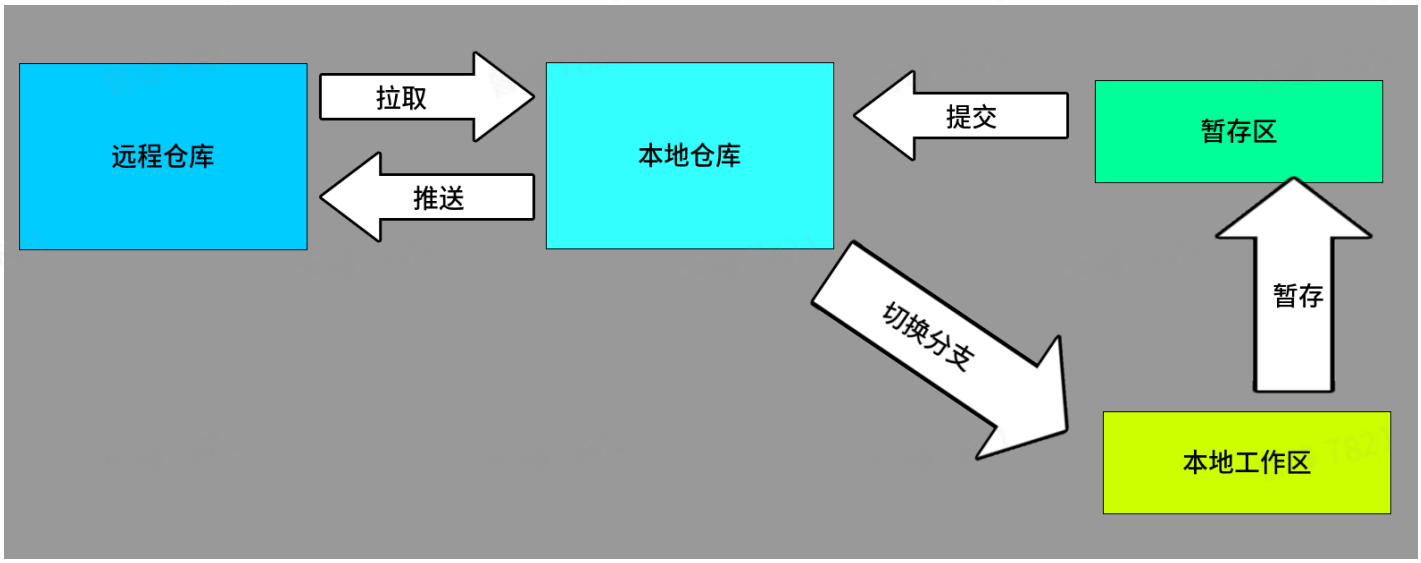


在SourceTree上配置私钥

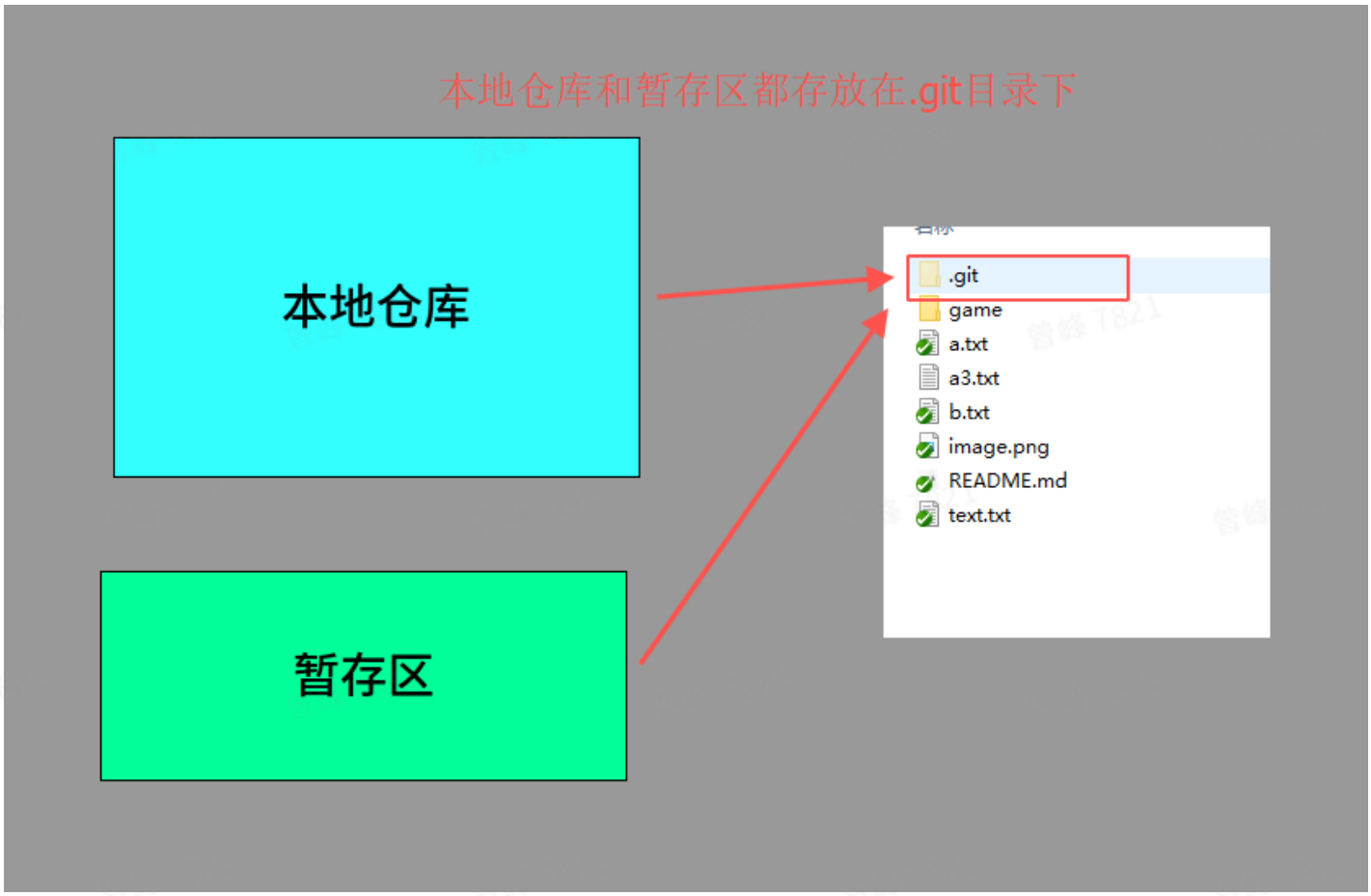



选项
一般
Updates
比较
Git
Mercurial
自定义操作
验证
网络
允许 Sourcetree 修改您的全局 Git 和 Mercurial 配置文件
Open links on Bitbucket.org with Sourcetree
在未来提议创建书签
Theme: Light
1
4.配置一下自己的用户名，方便区别谁提交的
默认用户信息
全名: zengfeng75
电子邮件地址: zengfeng75@gmail.com
SSH客户端配置
SSH 密钥: C:\Users\FengZeng\.ssh\id_rsa
SSH 客户端: OpenSSH
(仅针对 Git，Mercurial 总是使用 Plink on Windows)
PuTTY / Plink
OpenSSH
2
3. 做完步骤2自动会填
Repo Settings
项目目录:
语言: 自动
(需要重启) 帮助翻译 Sourcetree!
默认文本编码: utf-8
进行破坏性操作时保留备份
当文件有改动时自动刷新
5
确定

6. Git仓库关系介绍



本地仓库和暂存区都存放在.git目录下。

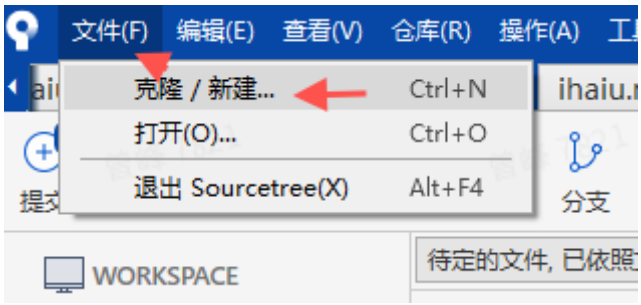


本地工作区就是除.git外的所有文件。

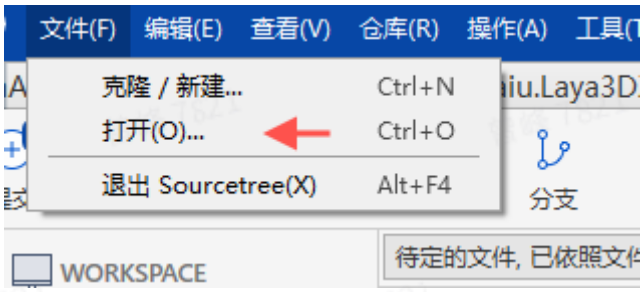
二：SourceTree 使用

https://blog.csdn.net/syk123839070/article/details/82534968
https://www.cnblogs.com/tian-xie/p/6264104.html

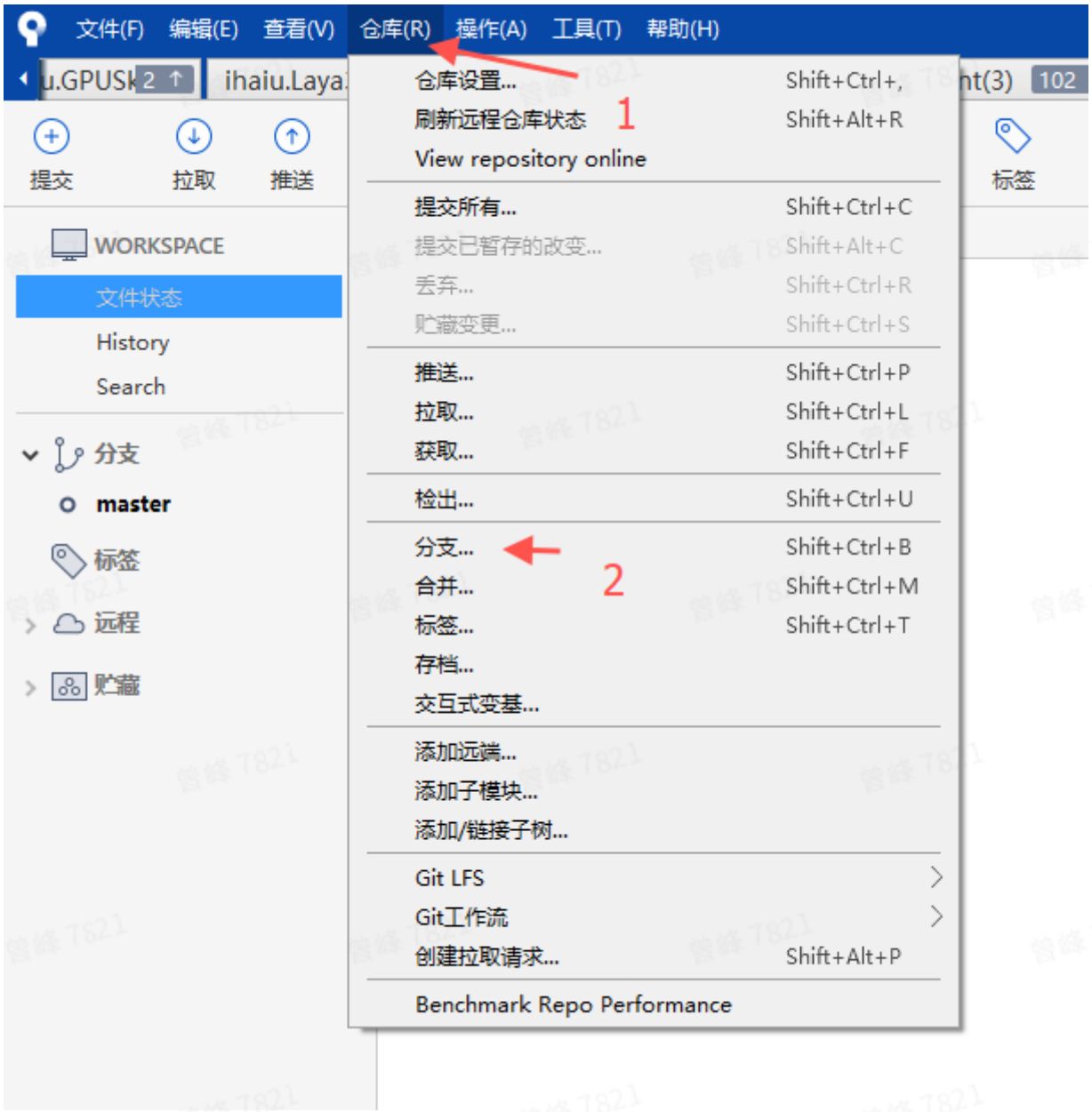
（1）克隆仓库到本地



（2）添加本地已经克隆过的仓库

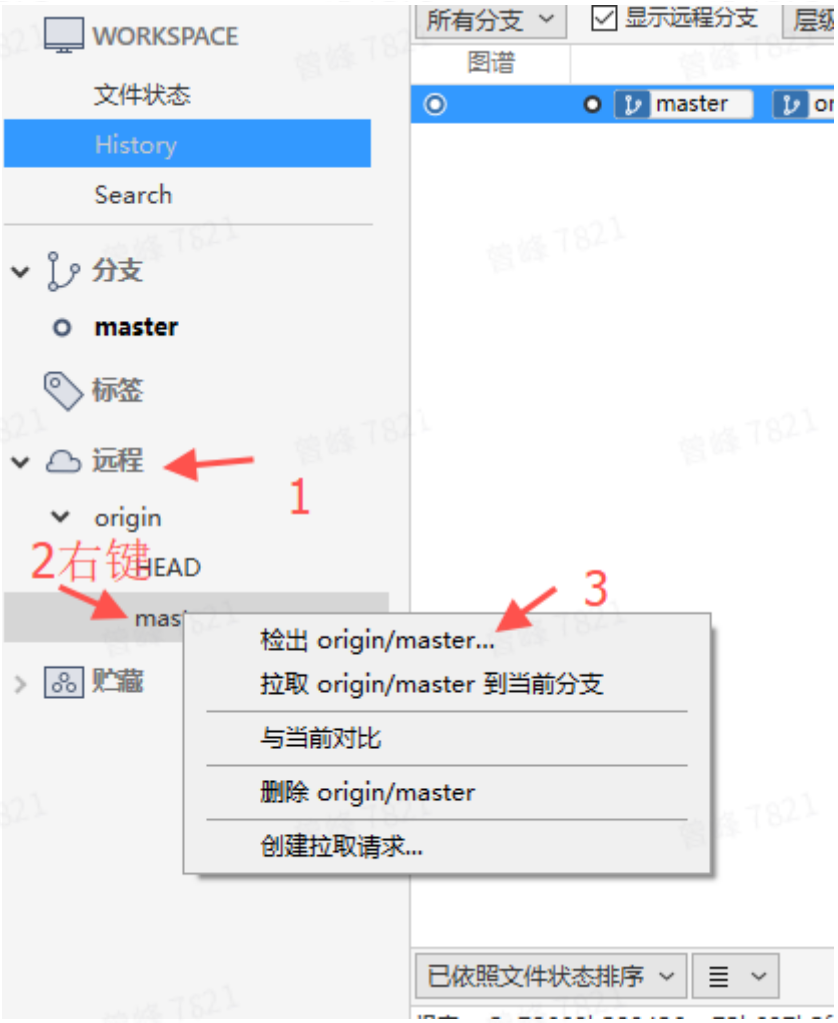


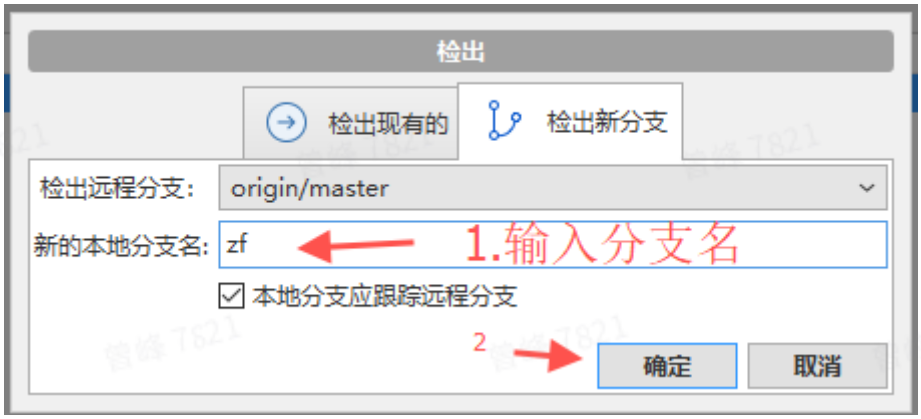
（3）创建分支--从当前本地分支


文件(F)
编辑(E)
查看(V)
仓库(R)
操作(A)
工具(T)
帮助(H)
提交
拉取
推送
标签
WORKSPACE
文件状态
History
Search
分支
master
标签
远程
贮藏
仓库设置... Shift+Ctrl+,
刷新远程仓库状态 Shift+Alt+R
1
View repository online
提交所有... Shift+Ctrl+C
提交已暂存的改变... Shift+Alt+C
丢弃... Shift+Ctrl+R
贮藏变更... Shift+Ctrl+S
推送... Shift+Ctrl+P
拉取... Shift+Ctrl+L
获取... Shift+Ctrl+F
检出... Shift+Ctrl+U
分支... Shift+Ctrl+B
合并... Shift+Ctrl+M
2
标签... Shift+Ctrl+T
存档...
交互式变基...
添加远端...
添加子模块...
添加/链接子树...
Git LFS
Git工作流
创建拉取请求... Shift+Alt+P
Benchmark Repo Performance

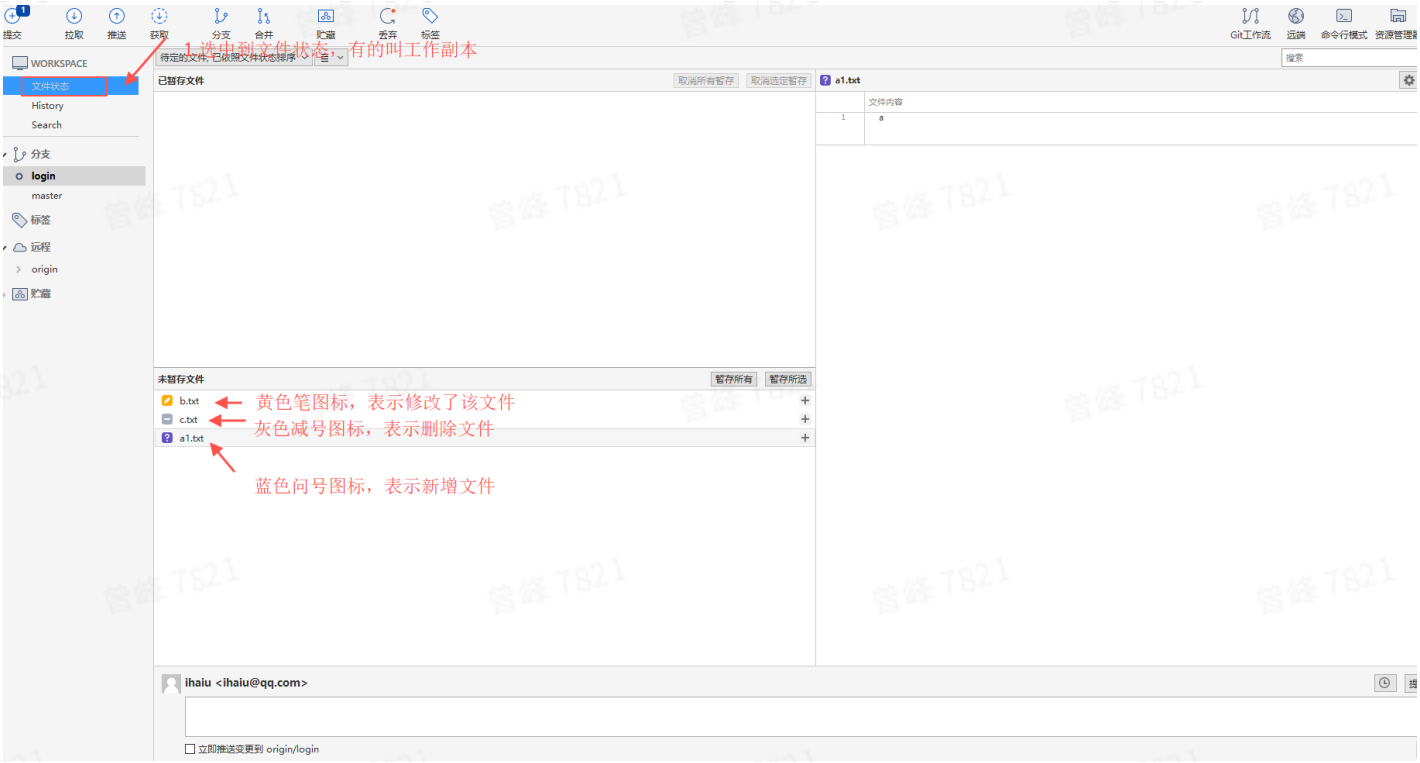

分支
新分支
删除分支
1.输入自己的分支名，确保目前没有相同的分支名
当前分支: master
新分支: login
提交:
工作副本原本
指定的提交:
...
检出新分支
2
创建分支
取消

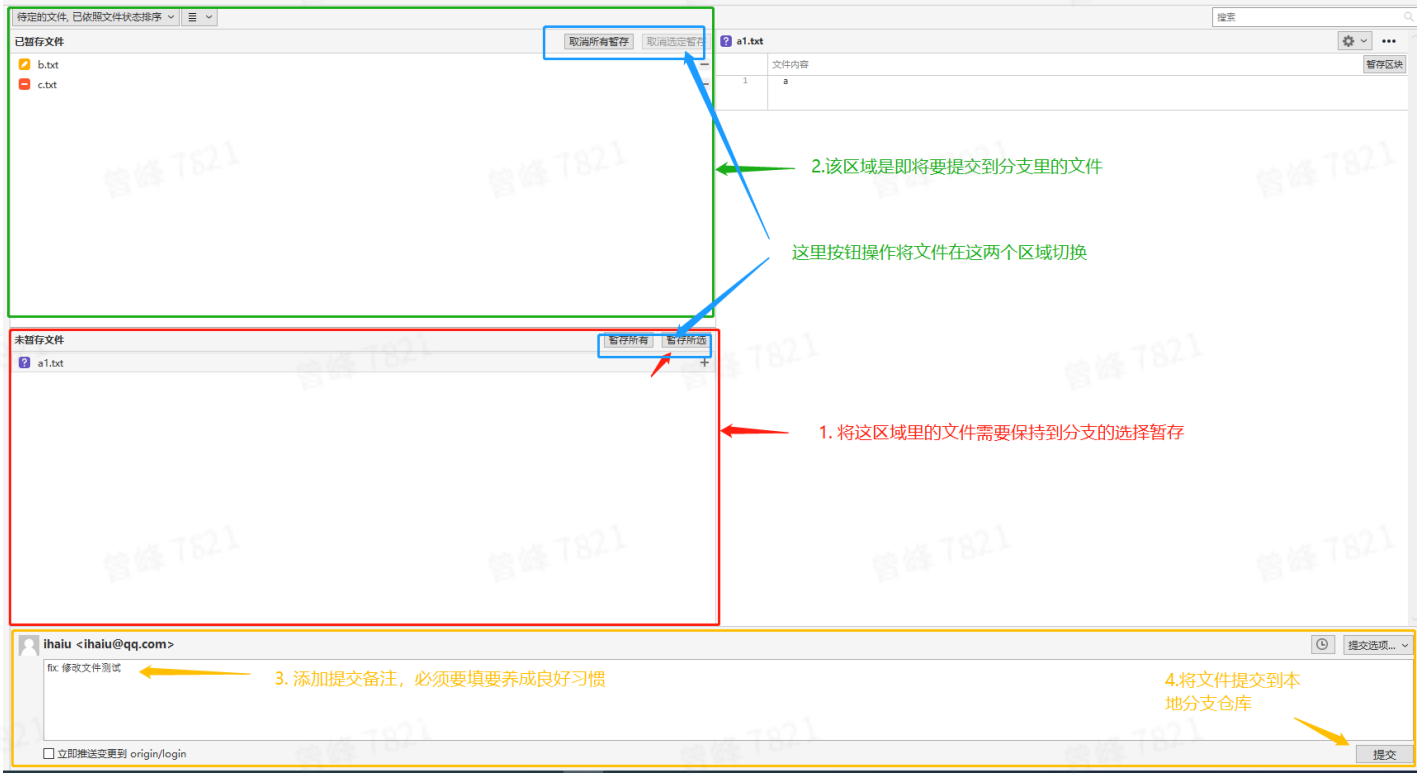
（4）创建分支--从远程分支


WORKSPACE
文件状态
History
Search
分支
master
标签
远程
origin
HEAD
贮藏
所有分支
显示远程分支
图谱
master
1
2右键
3
检出 origin/master...
拉取 origin/master 到当前分支
与当前对比
删除 origin/master
创建拉取请求...
已依照文件状态排序


检出
检出现有的
检出新分支
检出远程分支:
origin/master
新的本地分支名:
zf
1.输入分支名
本地分支应跟踪远程分支
2
确定
取消

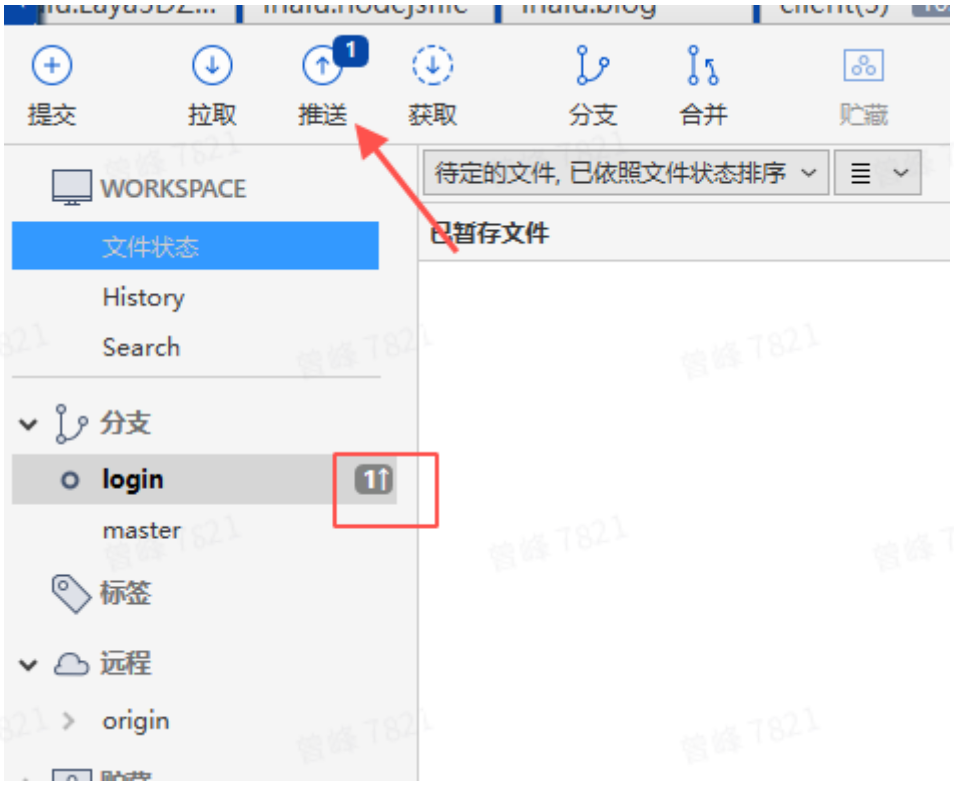
（5）提交文件





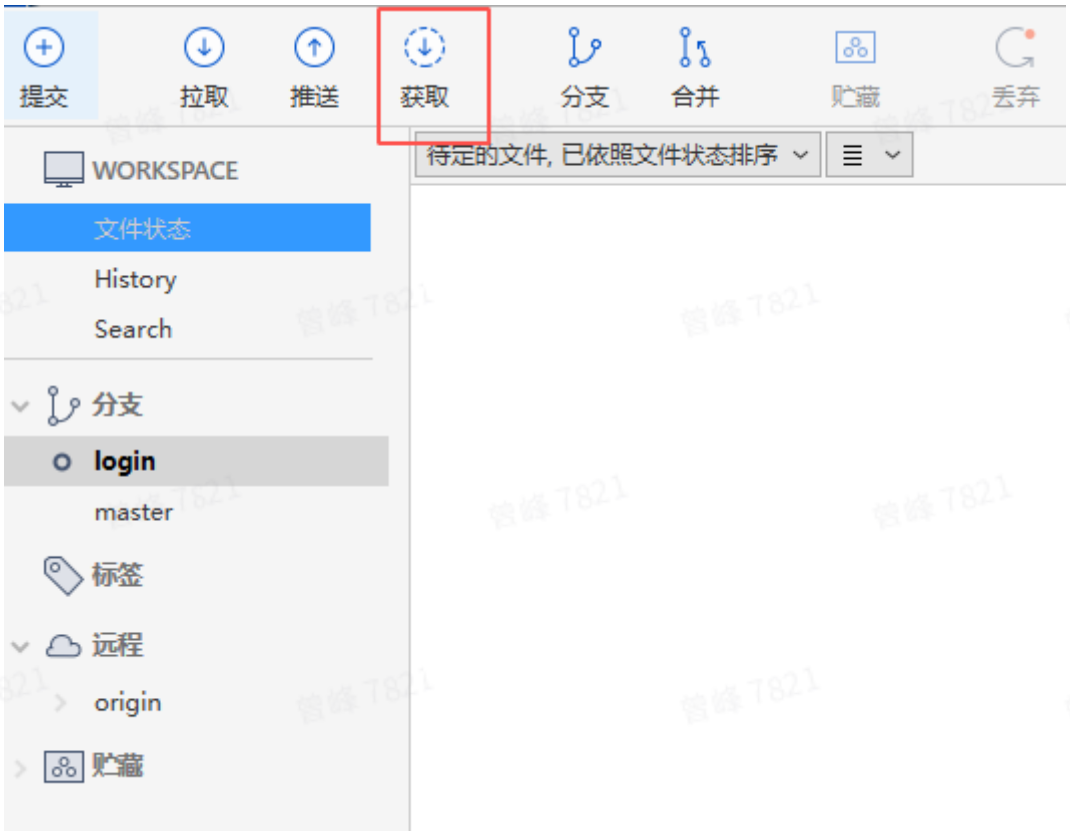
（5）推送分支

将本地分支，推送到远程仓库分支



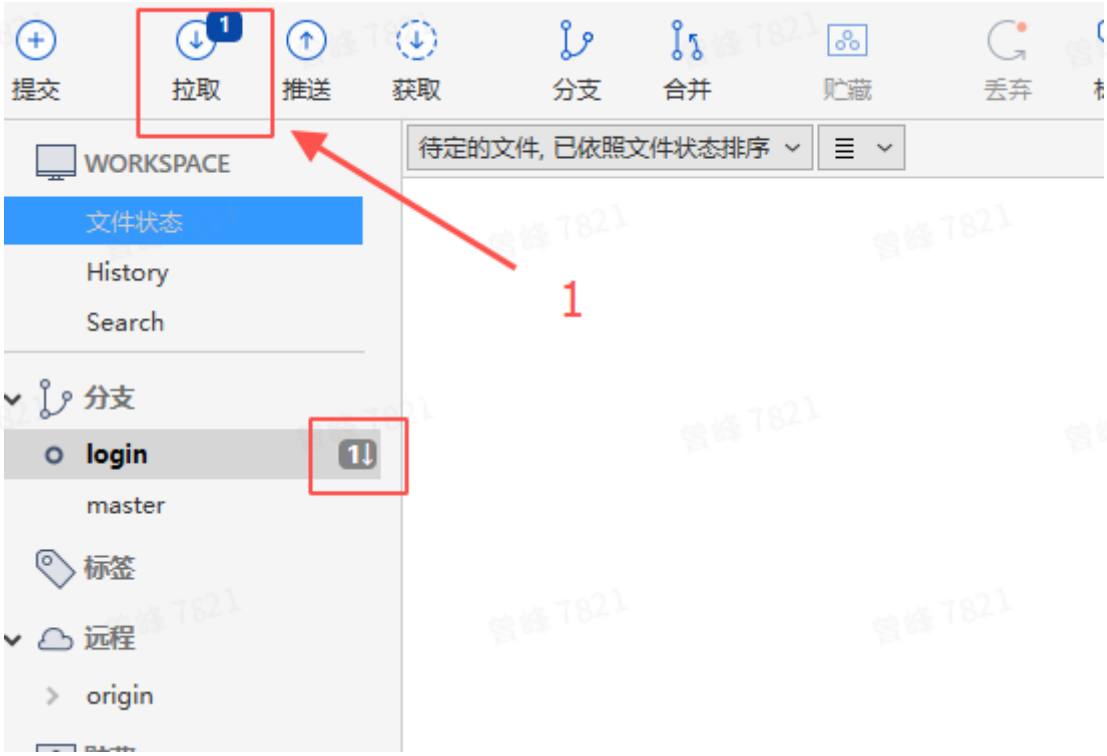
（6）获取远程仓库信息

可以获取其他开发人员提交到远程仓库的分支信息。不影响当前本地分支。

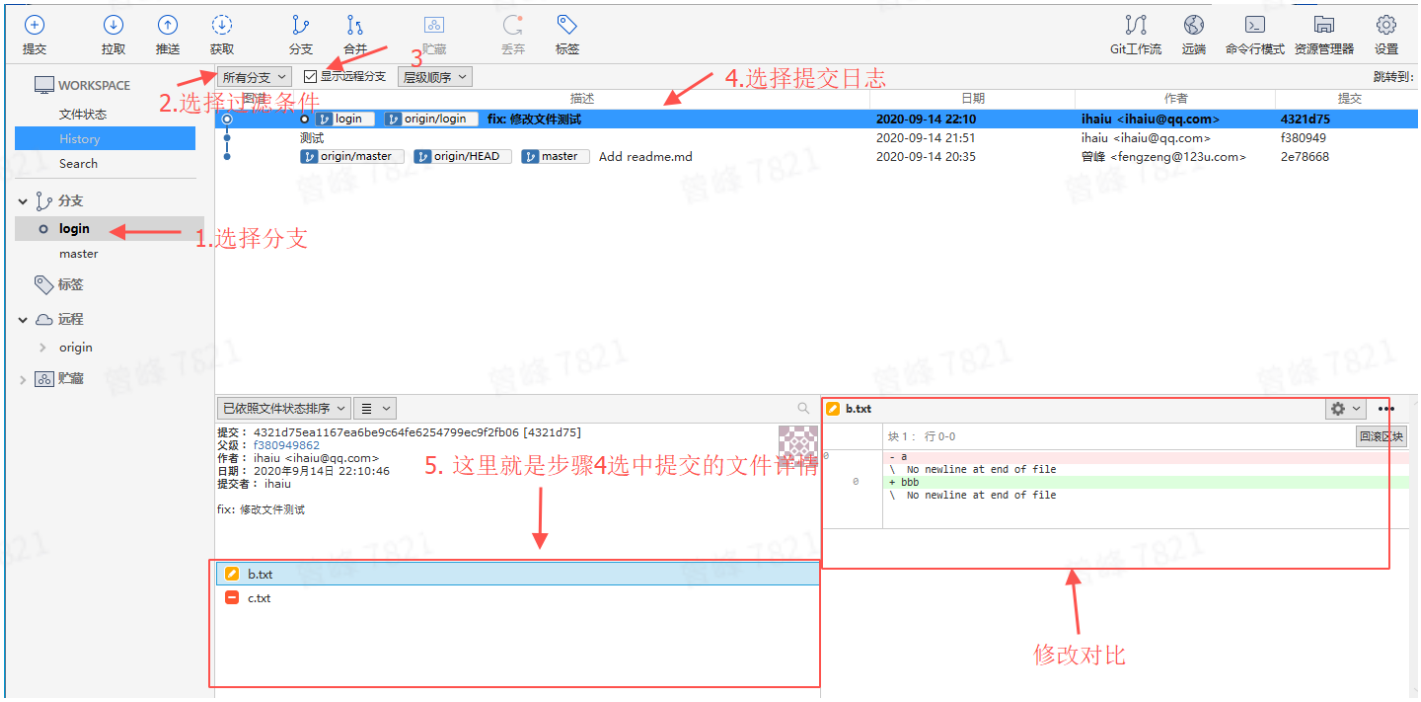


（7）拉取分支

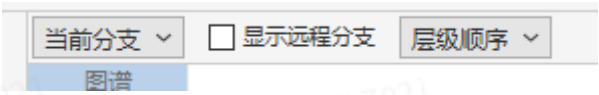
将远程分支修改同步到本地。



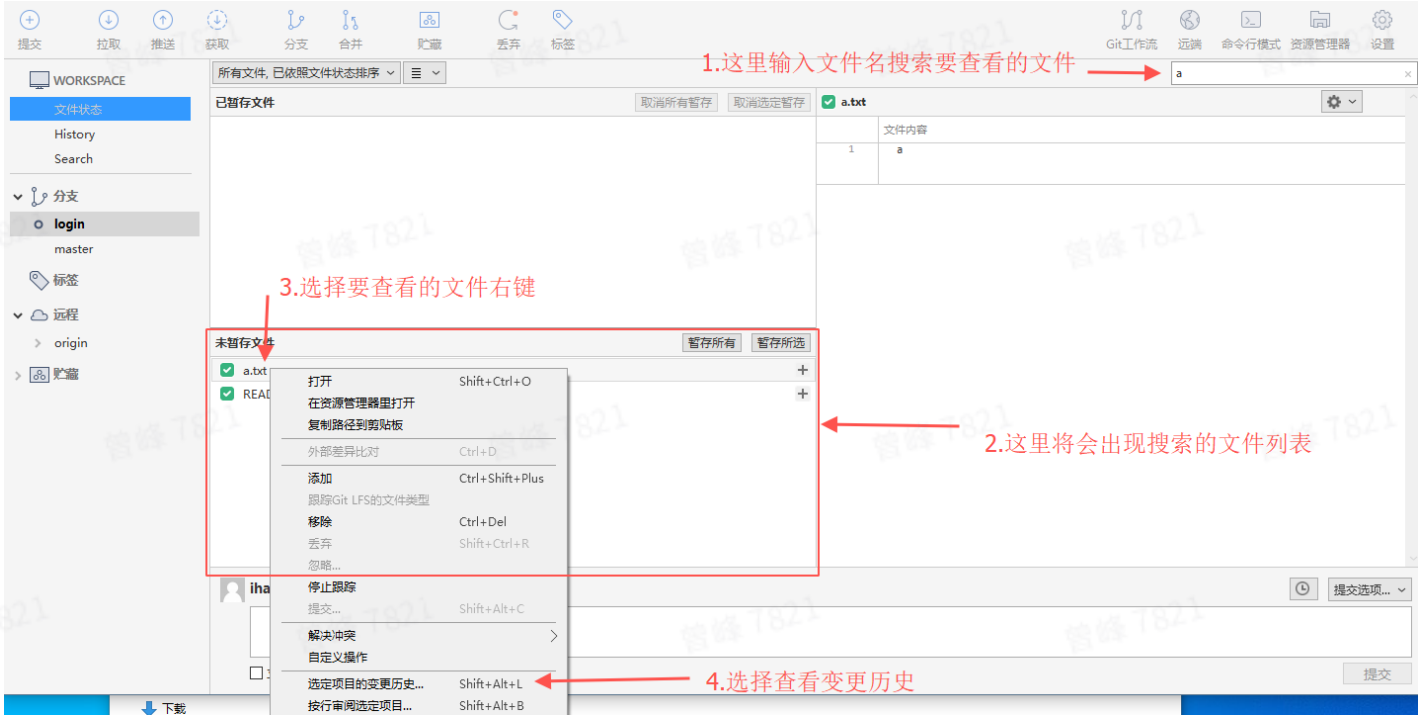
（8）查看提交日志

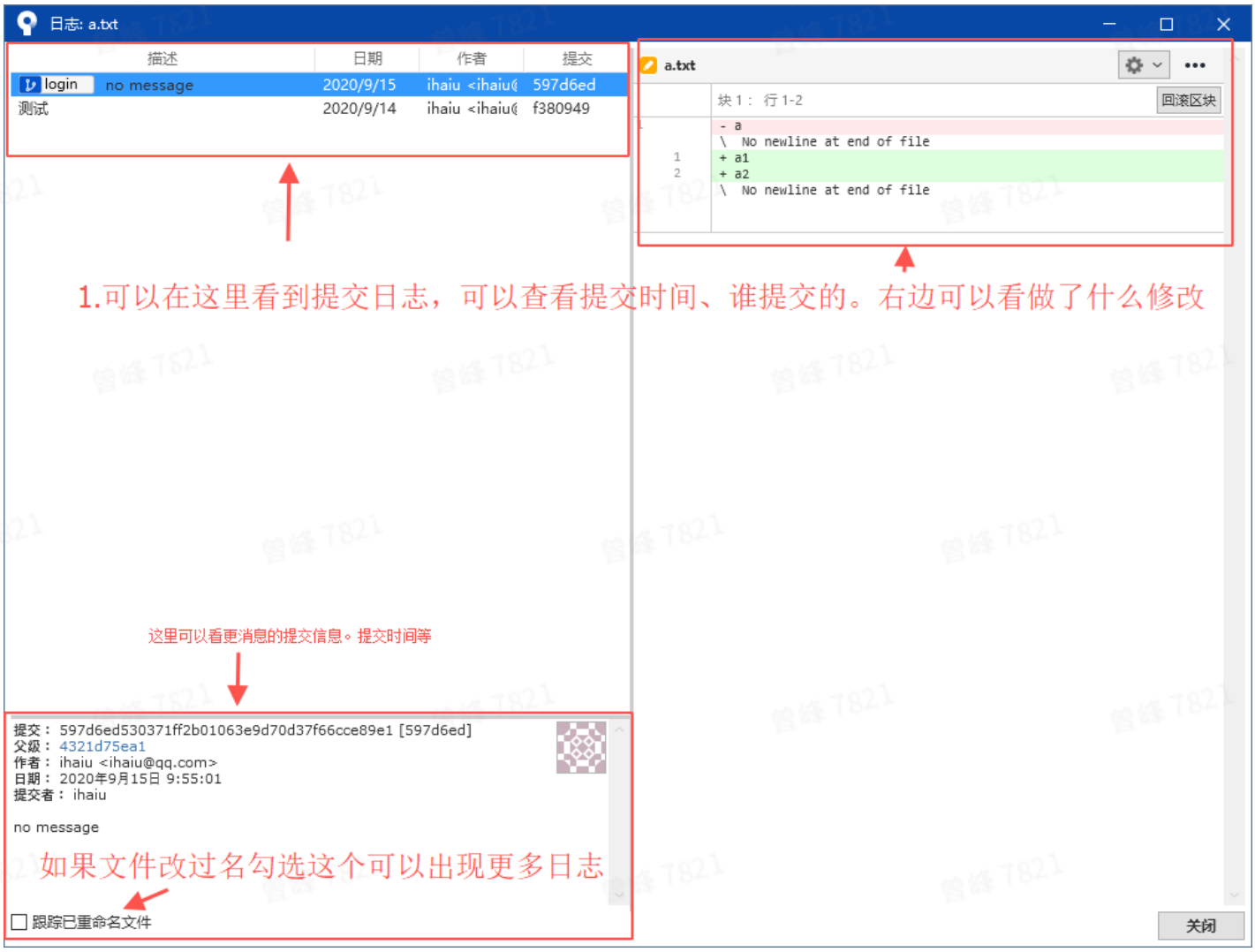


一般我们选择本地当前分支查看。

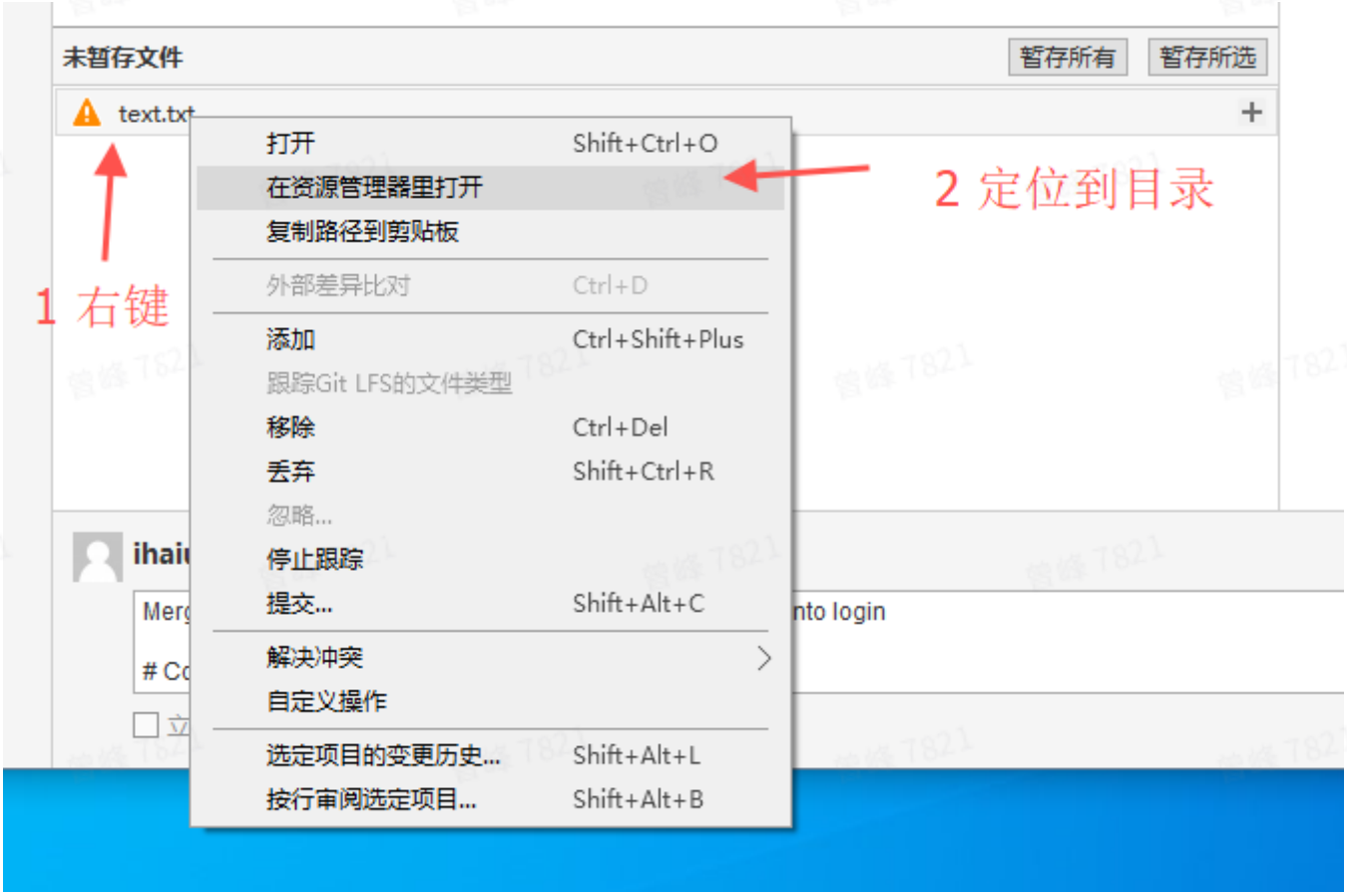


（9）查看文件历史记录




日志: a.txt
描述
日期
作者
提交
login
no message
2020/9/15
ihaiu <ihaiu
597d6ed
测试
2020/9/14
ihaiu <ihaiu
f380949
a.txt
块 1： 行 1-2
回滚区块
- a
\ No newline at end of file
+ a1
+ a2
\ No newline at end of file
1.可以在这里看到提交日志，可以查看提交时间、谁提交的。右边可以看做了什么修改
这里可以看更消息的提交信息。提交时间等
提交： 597d6ed530371ff2b01063e9d70d37f66cce89e1 [597d6ed]
父级： 4321d75ea1
作者： ihaiu <ihaiu@qq.com>
日期： 2020年9月15日 9:55:01
提交者： ihaiu
no message
如果文件改过名勾选这个可以出现更多日志
跟踪已重命名文件
关闭

（10）查看文件所在目录



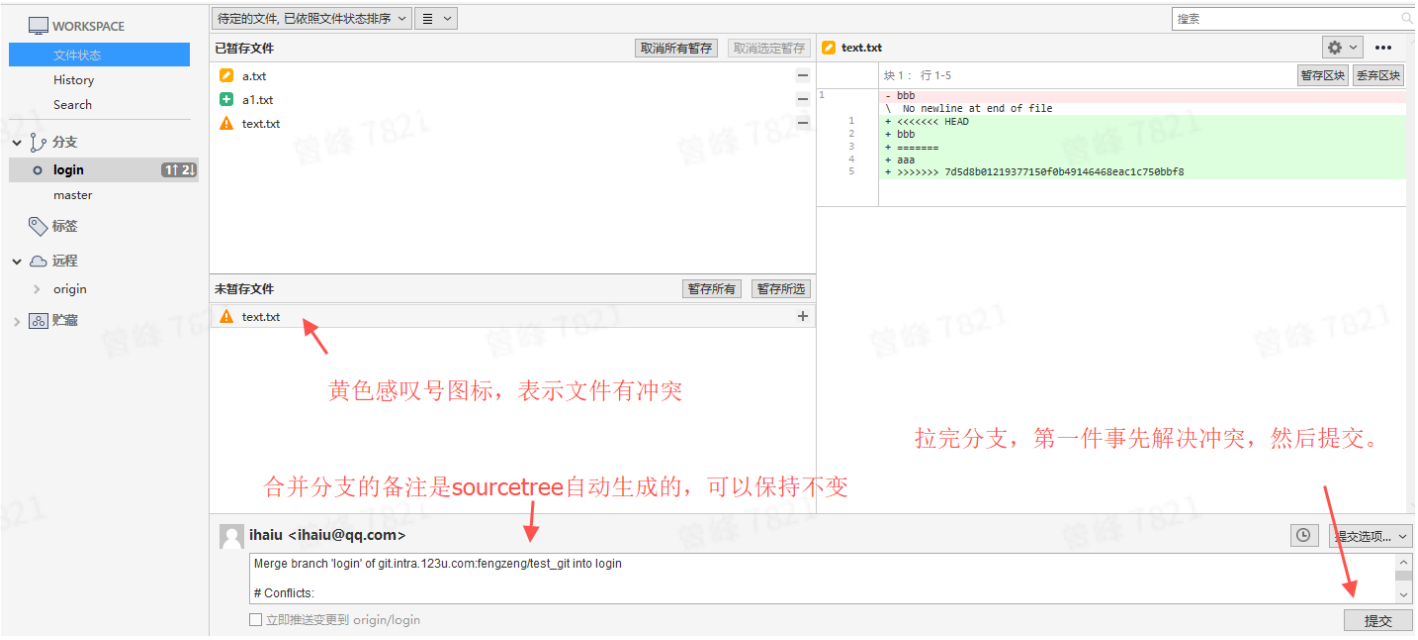
（11）文件冲突

冲突发生的原因：你和其他成员在这段时间有对同路径名称的文件都有操作。
解决的目的：让这文件满足你们双方需求

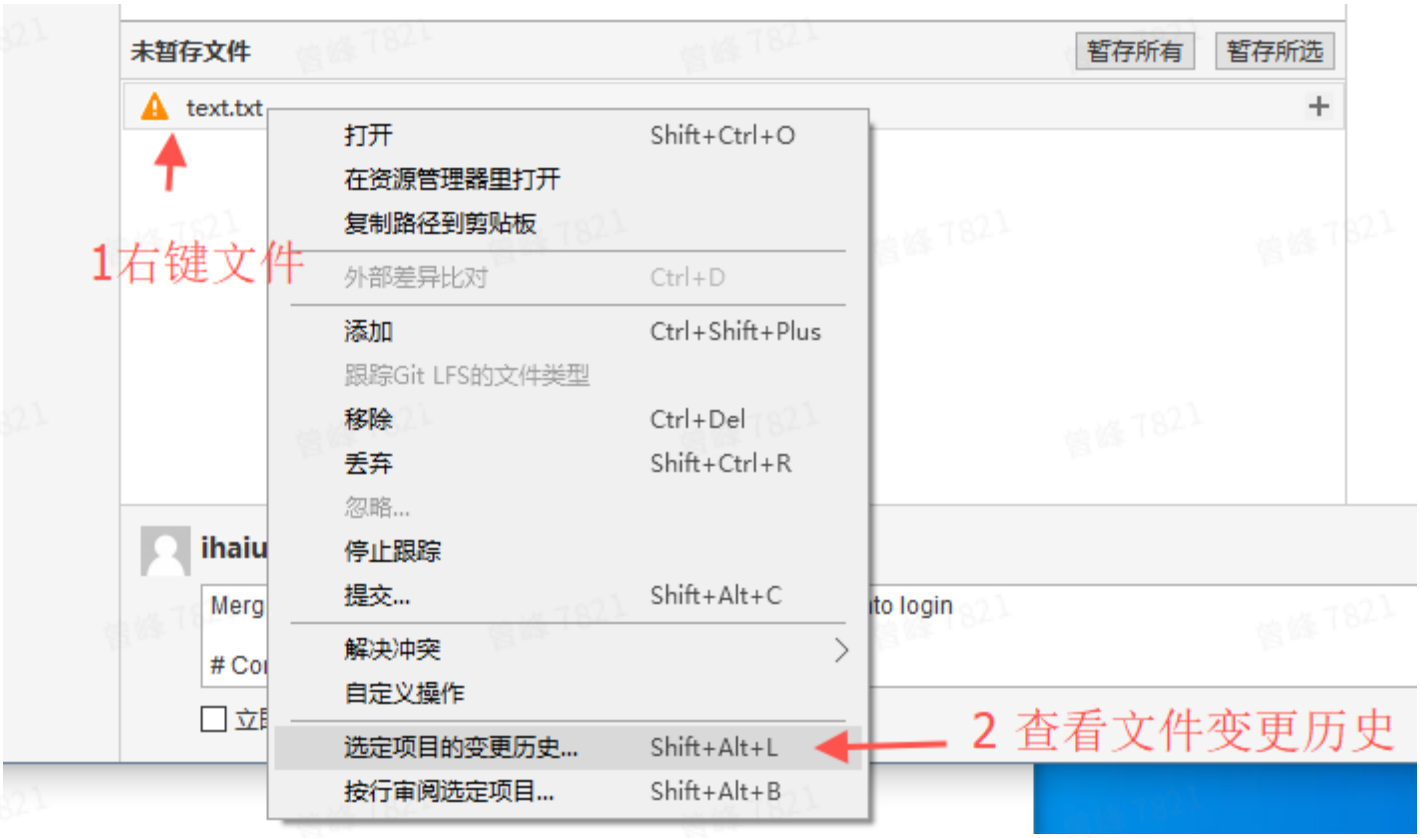
解决方法：
1.找最后一次提交该文件的同事询问。
2.你们双方协商使用谁的，或者怎么修改。

解决冲突流程

拉取分支，如果有冲突，将会出现黄色感叹号图标的文件。

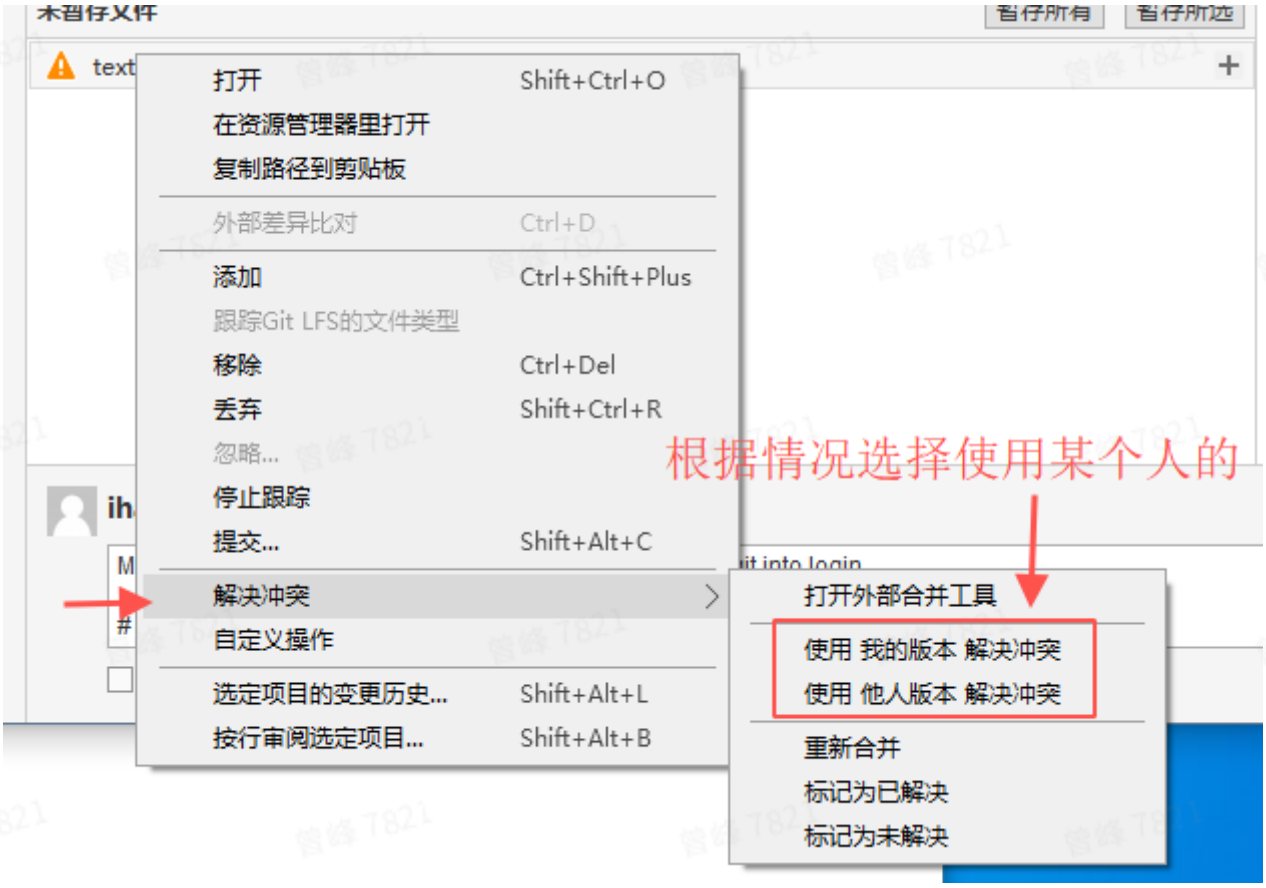


查看该文件提交日志, 找到最后一次提交该文件的作者，协商解决冲突



如果你们协商好了。一般有以下几种方式解决：

方式一： 选择使用某个人的

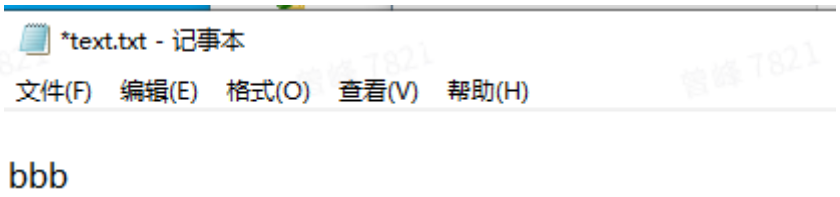


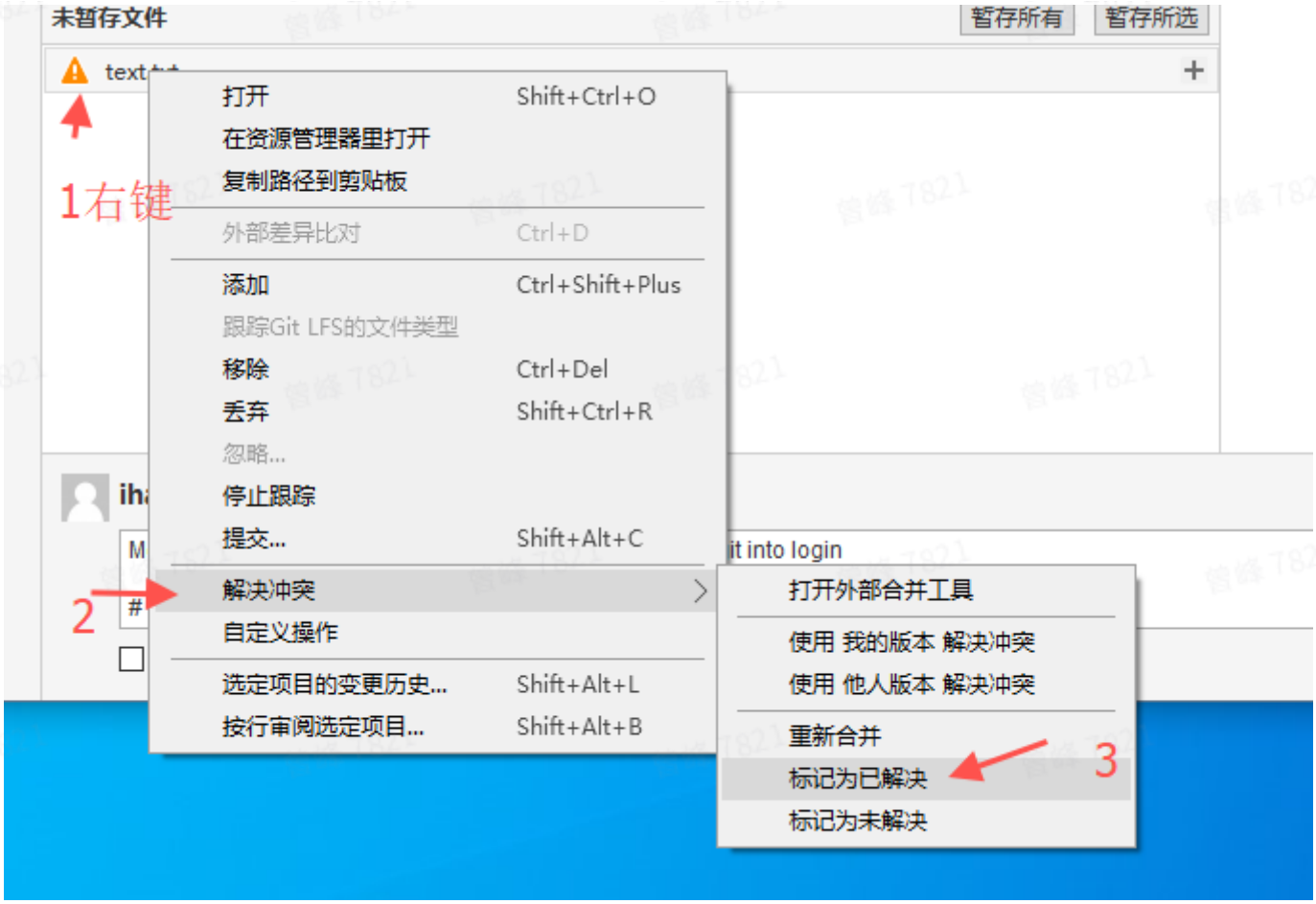
方式二： 先修改文件，然后标记为已解决

文本文件可以直接双击打开，其他文件需要定位到文件目录位置替换该文件。

文本文件：

根据情况修改文本。然后标记已修改。

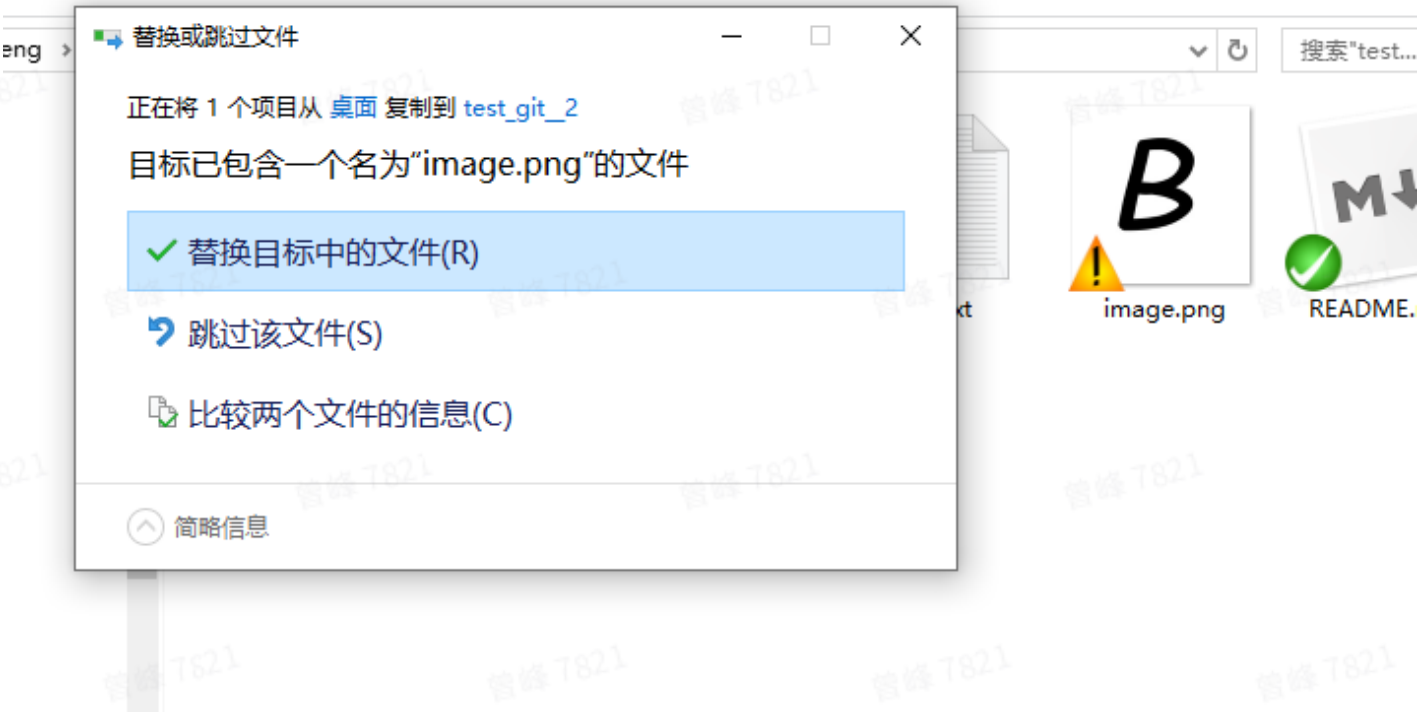


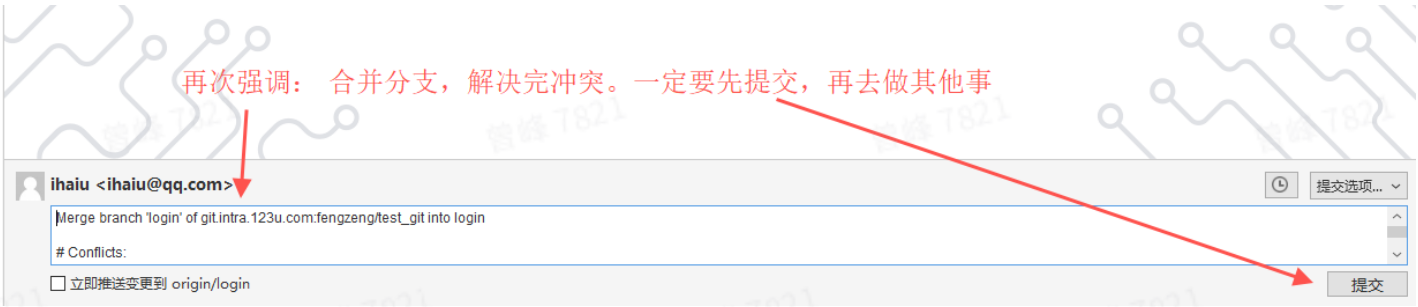


二进制文件，比如图片

先定位到图片所在目录

将自己在其他目录对应的文件复制粘贴，覆盖该文件。然后标记为已解决。

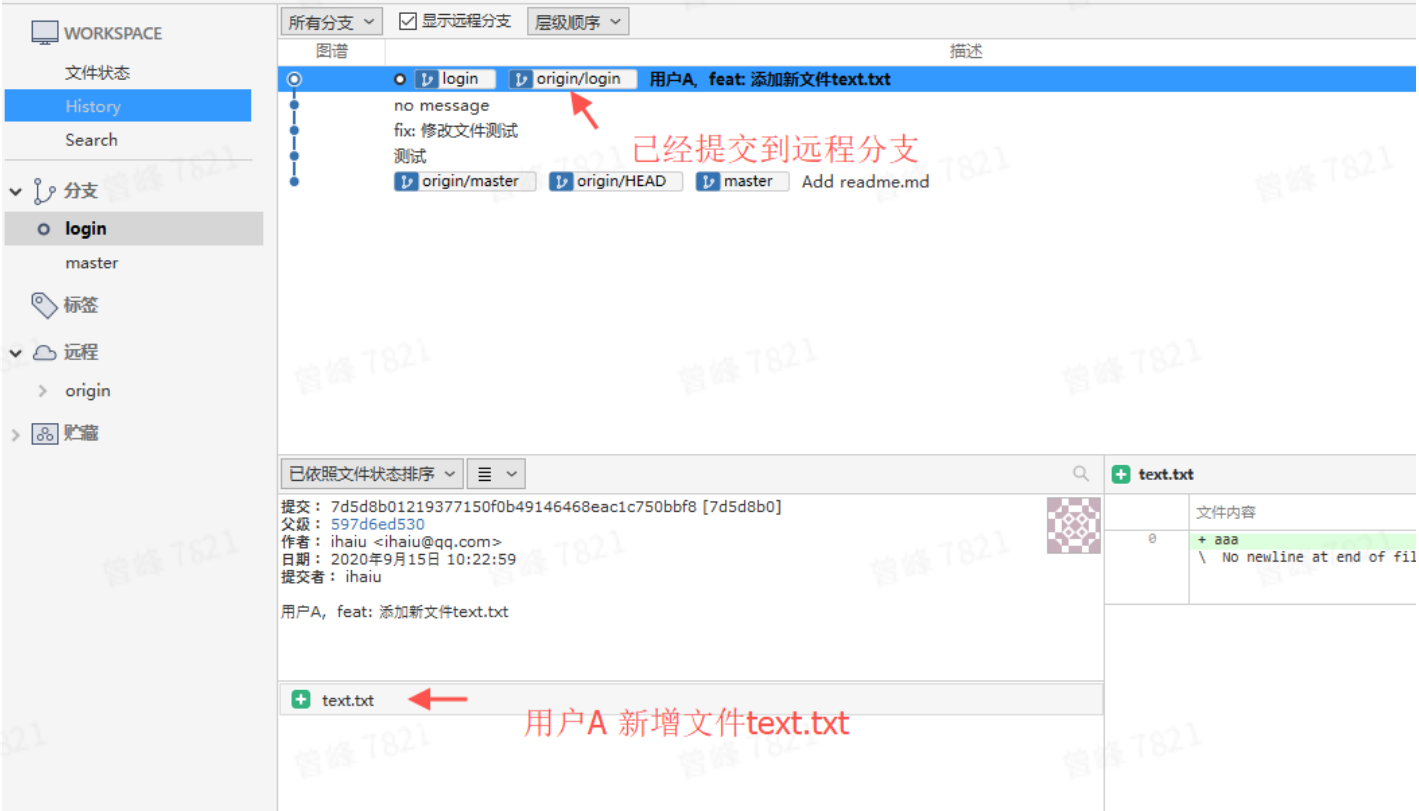


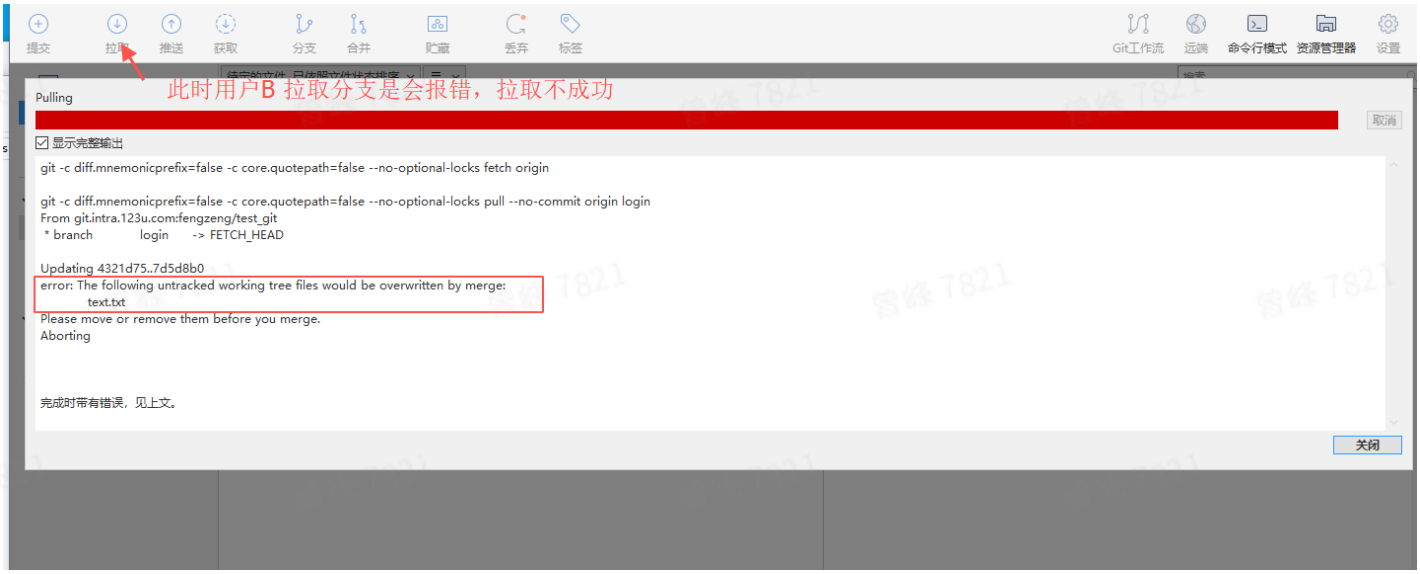


冲突情况一：用户同时新增相同路径名称文件

用户A： 新增文件， text.txt，提交到远程分支。

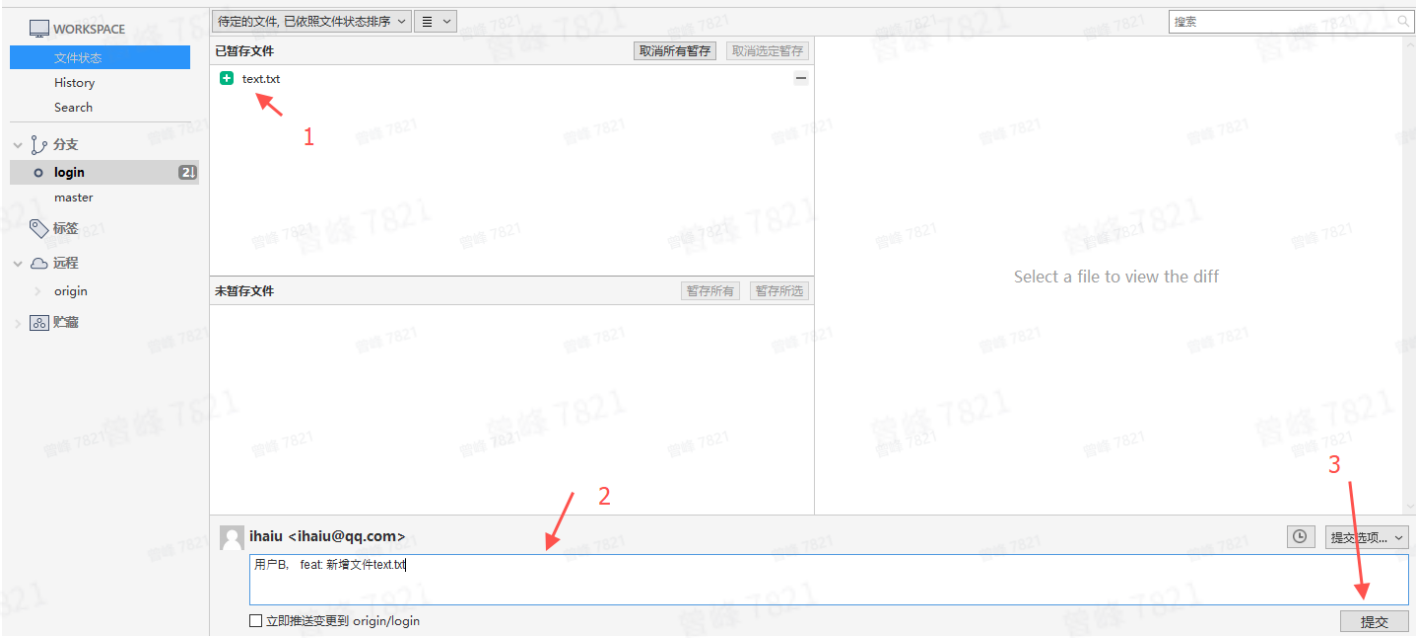
用户B： 新增文件， text.txt，拉取远程分支。



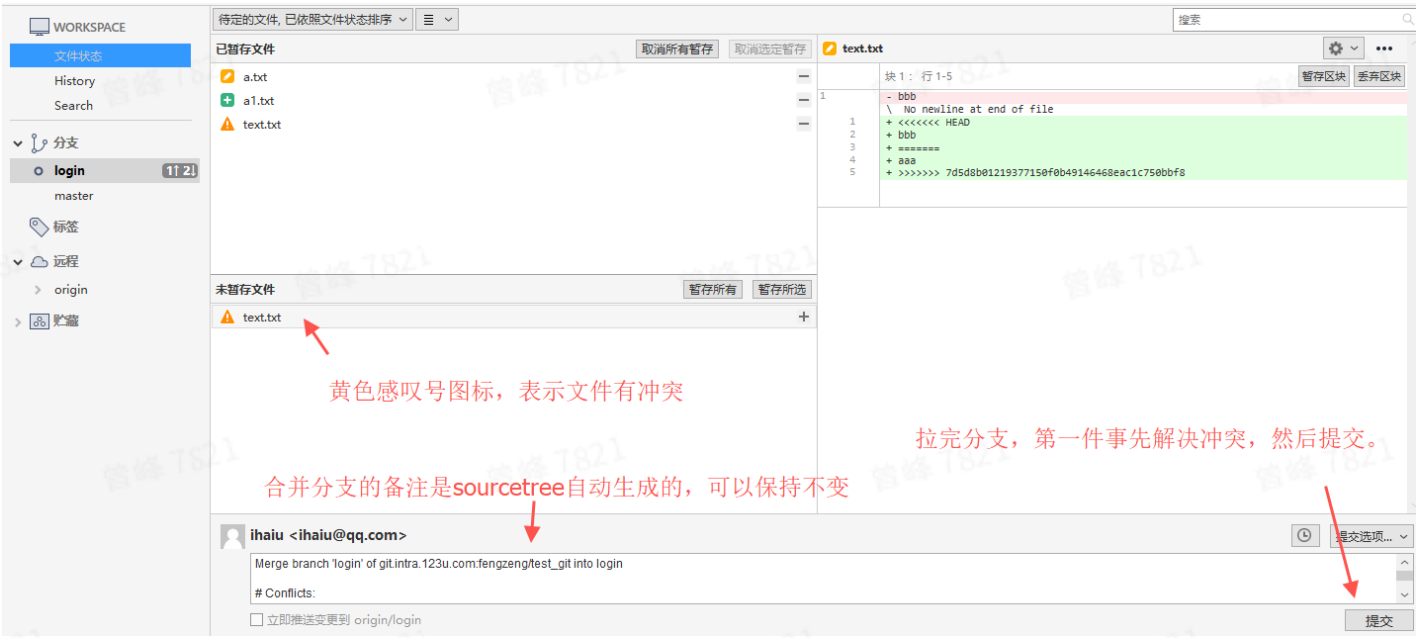


解决方法：

用户B，先将text.txt 提交本地分支



此时再拉取，可以成功把远程的拉到本地，但是将会出现冲突。



然后走”解决冲突流程“

冲突情况二：用户同时修改相同路径名称文件

避免方法：如果修改的不是自己文件，先和对方说一下，让对方先提交，你拉取完成再去修改。

（12）忽略文件

忽略文件只存放在本地文件夹中，不会进git仓库，更不会传到远程仓库。

原则：

1. 合并分支时，不准"忽略"和"停止跟踪"别人的文件。
2. 只能忽略自己文件，并且是确保分支中没有依赖该文件。

（13）停止跟踪文件

停止跟踪是将仓库中已经存在的文件删除。

原则：

1. 合并分支时，不准"忽略"和"停止跟踪"别人的文件。
2. 不允许美术使用

（14）丢弃文件

丢弃文件是将还未暂存的文件，放弃操作（修改、删除）。

原则：

1. 只允许对 黄色笔图标 修改的文件使用

（14）移除文件

移除文件是删除本地文件。如果是“黄色笔图标的修改文件“将会改变为“灰色减号图标的删除文件”，如果将该文件继续提交将会删除仓库文件，再推送将会删除远程仓库文件。

原则：

1. 不允许乱移除其他人的文件。不是自己的文件需要和对方协商。
2. 美术只能移除问号图标新增文件

三：Git 使用规范

1. 拉取完分支：先解决冲突 -> 提交 -> 再去做自己的事。 否则自己的修改将会和合并分支日志混杂在一起，会很混乱。

2. 如果遇到自己不会解决的冲突，先查看日志，找到最后提交该文件的人协商。双方都不会解决找程序去帮忙。
3. 如果要动别人的文件，一定要事先和对方协商。否则不准动。
4. 避免冲突的方法：不要去动别人的文件。如果要动，先叫对方提交->你拉取->然后再修改
5. 不要乱忽略文件，只有确保是自己的文件，且提交的文件中没有依赖该文件才能添加忽略。
6. 不允许"忽略"、"停止跟踪"、“移除”别人文件，否则将会使别人的提交丢失。

git commit 提交描述规范

https://www.jianshu.com/p/b584152d2a16

简单规范： [提交人] type: <描述>

type 是提交的类型，一般常用的如下：

- feat：新功能
- fix：修正
- refactor：重构
- chore：维护
- style：修正代码格式，无代码功能上的变化
- docs：文档的变更
- test: 测试