目录

**[1. git 提交后撤销提交到缓存区](#_Toc4307_WPSOffice_Level1)** **[2](#_Toc4307_WPSOffice_Level1)**

**[2. git 撤销缓存区到工作区](#_Toc3630_WPSOffice_Level1)** **[2](#_Toc3630_WPSOffice_Level1)**

**[3. git 撤销工作区的修改](#_Toc25282_WPSOffice_Level1)** **[2](#_Toc25282_WPSOffice_Level1)**

**[4. 删除文件命令](#_Toc3602_WPSOffice_Level1)** **[2](#_Toc3602_WPSOffice_Level1)**

**[5. git 跟踪删除文件](#_Toc15980_WPSOffice_Level1)** **[2](#_Toc15980_WPSOffice_Level1)**

[1. rm file后](#_Toc3630_WPSOffice_Level2) [2](#_Toc3630_WPSOffice_Level2)

[2. git rm file](#_Toc25282_WPSOffice_Level2) [2](#_Toc25282_WPSOffice_Level2)

[3. git commit –m ‘rm file’](#_Toc3602_WPSOffice_Level2) [2](#_Toc3602_WPSOffice_Level2)

**[6. 生成SHH Key](#_Toc20014_WPSOffice_Level1)** **[3](#_Toc20014_WPSOffice_Level1)**

**[7. 添加远程仓库](#_Toc21861_WPSOffice_Level1)** **[3](#_Toc21861_WPSOffice_Level1)**

**[8. 把本地提交内容推送到远程服务器](#_Toc4477_WPSOffice_Level1)** **[3](#_Toc4477_WPSOffice_Level1)**

**[9. 新建项目同步到（提交）远程失败](#_Toc18542_WPSOffice_Level1)** **[3](#_Toc18542_WPSOffice_Level1)**

[一、问题分析](#_Toc15980_WPSOffice_Level2) [3](#_Toc15980_WPSOffice_Level2)

[二、解决问题](#_Toc20014_WPSOffice_Level2) [4](#_Toc20014_WPSOffice_Level2)

**[10. 创建分支](#_Toc163_WPSOffice_Level1)** **[4](#_Toc163_WPSOffice_Level1)**

**[11. 删除分支](#_Toc9433_WPSOffice_Level1)** **[4](#_Toc9433_WPSOffice_Level1)**

**[12. 合并分支](#_Toc26082_WPSOffice_Level1)** **[5](#_Toc26082_WPSOffice_Level1)**

**[13. git用命令行实现从ssh切换到https](#_Toc7481_WPSOffice_Level1)** **[5](#_Toc7481_WPSOffice_Level1)**

[1、从ssh切换至https](#_Toc21861_WPSOffice_Level2) [5](#_Toc21861_WPSOffice_Level2)

[2、从https切换至ssh](#_Toc4477_WPSOffice_Level2) [5](#_Toc4477_WPSOffice_Level2)

[3、查看当前是ssh还是https](#_Toc18542_WPSOffice_Level2) [5](#_Toc18542_WPSOffice_Level2)

**[14. 安装好SourceTree后，首次使用时，需要通过账户登录，但注册或登录界面可能根本无法打开，导致无法进入软件](#_Toc15758_WPSOffice_Level1)** **[6](#_Toc15758_WPSOffice_Level1)**

[解决办法](#_Toc163_WPSOffice_Level2) [6](#_Toc163_WPSOffice_Level2)

**[15. 配置sourcetree](#_Toc10352_WPSOffice_Level1)** **[8](#_Toc10352_WPSOffice_Level1)**

**[16. Git本地分支和远程分支如何关联](#_Toc26461_WPSOffice_Level1)** **[8](#_Toc26461_WPSOffice_Level1)**

## git 提交后撤销提交到缓存区

* 1. 如果保留修改的撤销->只删掉提交记录<- ： git reset HEAD^ 或者 git reset XXX(上一次哈希值)
  2. $ git reset --soft 5029f0cc08cf（上一次哈希值） //保修修改
  3. 如果把修改记录也干掉，用git reset --hard HEAD^

## git 撤销缓存区到工作区

可以把暂存区的修改撤销掉（unstage），重新放回工作区

file(要撤销修改的文件名如readme.txt)

git reset HEAD file

## git 撤销工作区的修改

file(要撤销修改的文件名)

git checkout – file

## 删除文件命令

rm file

## git 跟踪删除文件

* 1. rm file后
  2. git rm file
  3. git commit –m ‘rm file’

## 生成SHH Key

ssh-keygen -t rsa -C "897380685@qq.com"

## 添加远程仓库

git remote add origin git@github.com:michaelliao/learngit.git

## 把本地提交内容推送到远程服务器

git push -u origin master

由于远程库是空的，我们第一次推送master分支时，加上了-u参数，Git不但会把本地的master分支内容推送的远程新的master分支，还会把本地的master分支和远程的master分支关联起来，在以后的推送或者拉取时就可以简化命令。

## 新建项目同步到（提交）远程失败

wbf[@wangbaofeng](https://my.oschina.net/wangbaofeng" \t "https://my.oschina.net/wangbaofeng/blog/_blank) MINGW64 /e/gitProjects/myDemoPro (master)  
$ git push origin master  
To git@github.com:zuiwangzuo/myDemoPro.git  
 ! [rejected]        master -> master (fetch first)  
error: failed to push some refs to 'git@github.com:zuiwangzuo/myDemoPro.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.

****一、问题分析****

先到github上看看远程repository下到底有啥东西，发现目标库是空的。

但是有一个README.md文件（本地代码中没有），而出现错误的原因是github中的README.md文件不在本地代码目录中。

也就是说我们需要先将远程代码库中的任何文件先pull到本地代码库中，才能push新的代码到github代码库中。

****二、解决问题****

1. 将远程代码库中的代码pull到本地代码库，并且执行合并merge：

$ git pull --rebase origin master  
warning: no common commits  
remote: Counting objects: 3, done.  
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0  
Unpacking objects: 100% (3/3), done.  
From github.com:zuiwangzuo/myDemoPro  
 \* branch            master     -> FETCH\_HEAD  
 \* [new branch]      master     -> origin/master  
First, rewinding head to replay your work on top of it...  
Applying: 提交初始代码。

2. 再次push代码：

$ git push origin master  
Counting objects: 20, done.  
Delta compression using up to 2 threads.  
Compressing objects: 100% (12/12), done.  
Writing objects: 100% (20/20), 2.51 KiB | 0 bytes/s, done.  
Total 20 (delta 0), reused 0 (delta 0)  
To git@github.com:zuiwangzuo/myDemoPro.git  
   8227f7c..aff8663  master -> master

我们看到本地代码已经成功的提交到远程代码中，至此问题解决。

## 创建分支

git checkout –b dev === git branch dev + git checkout dev

## 删除分支

git branch –d dev

## 合并分支

git merge dev

解决冲突

修改完冲突行后重新提交

## git用命令行实现从ssh切换到https

1、从ssh切换至https

git remote set-url origin(远程仓库名称) https://email/username/ProjectName.git

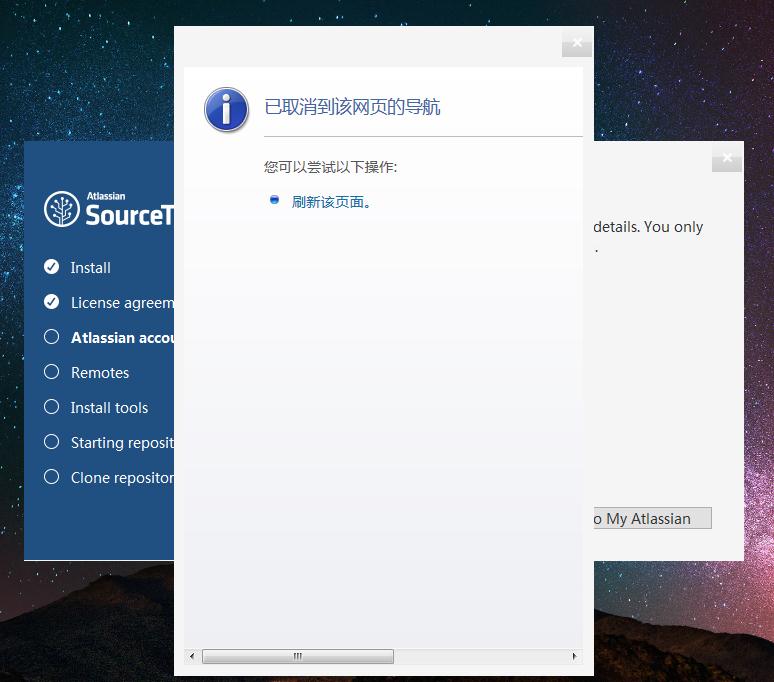
2、从https切换至ssh

git remote set-url origin git@email:username/ProjectName.git

3、查看当前是ssh还是https

git remote -v

## 安装好SourceTree后，首次使用时，需要通过账户登录，但注册或登录界面可能根本无法打开，导致无法进入软件



## 解决办法

在目录C:\Users\{youruser}\AppData\Local\Atlassian\SourceTree 下创建文件accounts.json ，注意：{youruser}需要替换为登录系统用户名。

写入如下内容：

[

{

"$id": "1",

"$type": "SourceTree.Api.Host.Identity.Model.IdentityAccount, SourceTree.Api.Host.Identity",

"Authenticate": true,

"HostInstance": {

"$id": "2",

"$type": "SourceTree.Host.Atlassianaccount.AtlassianAccountInstance, SourceTree.Host.AtlassianAccount",

"Host": {

"$id": "3",

"$type": "SourceTree.Host.Atlassianaccount.AtlassianAccountHost, SourceTree.Host.AtlassianAccount",

"Id": "atlassian account"

},

"BaseUrl": "https://id.atlassian.com/"

},

"Credentials": {

"$id": "4",

"$type": "SourceTree.Model.BasicAuthCredentials, SourceTree.Api.Account",

"Username": "username@email.com"

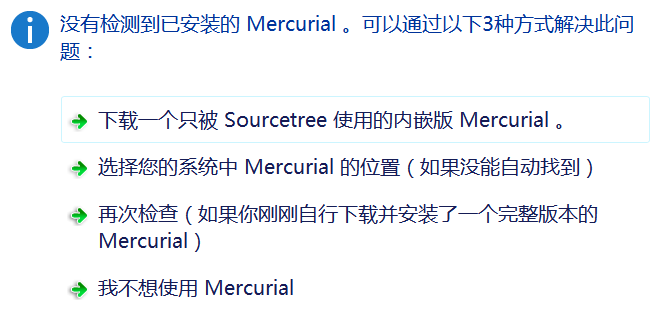
},

"IsDefault": false

}

]

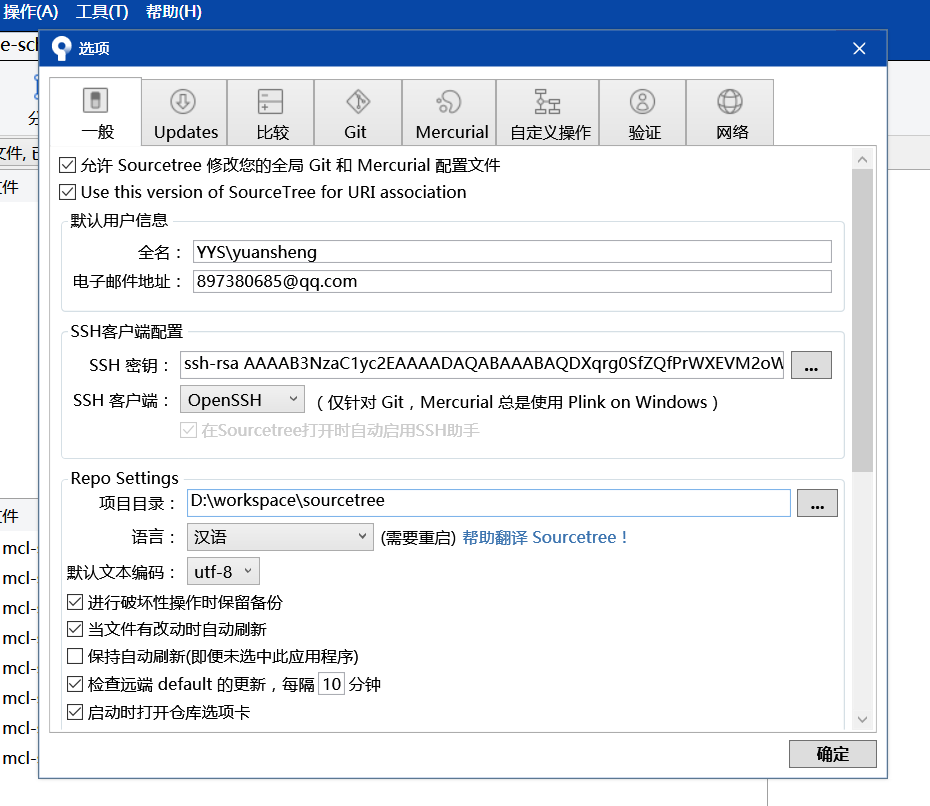
重新启动软件，顺利进入界面 ，如果有一下提示：



请到Mercurial官网下载安装：<https://www.mercurial-scm.org/>

## 配置sourcetree

参考网址：<https://www.cnblogs.com/wdlhao/p/9022511.html>



## Git本地分支和远程分支如何关联

****总结：****

****一般我们就用**git push --set-upstream origin branch\_name**来在远程创建一个与本地** branch\_name**同名的分支并跟踪；****

****利用**git checkout --track origin/branch\_name**来在本地创建一个与**branch\_name**同名分支 跟踪远程分支。****