

# Vscode配置Git+快速入门，一篇学会80%的Git操作

原创

小白还在写...

已于 2022-11-02 18:24:43 修改

10596 73

收藏

版权

分类专栏：


白马程序员

文章标签：

vscode

git

github

 白马程序员 专栏收录该内容

1 订阅 1 篇文章

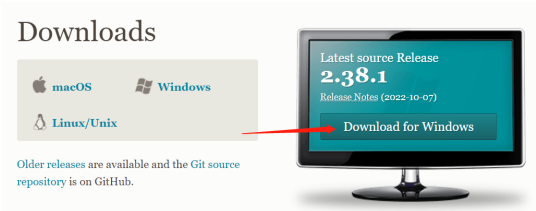
订阅专栏

## 前言

团队开发中经常会用到Git，能极大简化开发的流程，而个人开发也可以利用Git管理自己的代码。同样作为一个初学者，我在学完Git之后写下这篇文章总结个人走过的坑，大家一起进步。

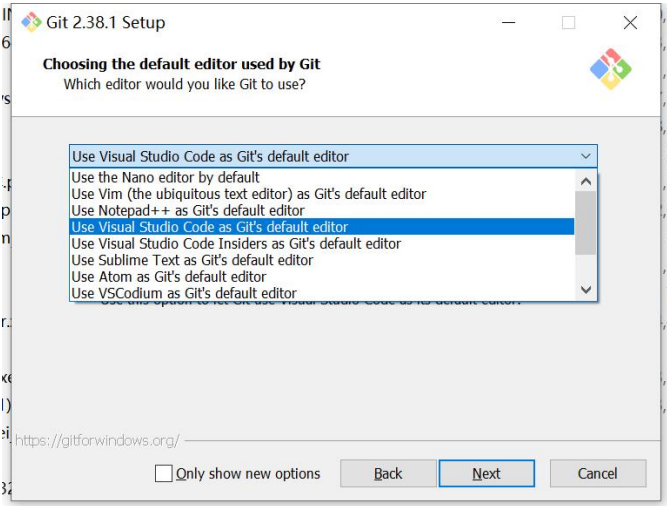
## Git下载和安装

下载地址：<https://git-scm.com/download>



点击下载64 Git for Windows Setup.

由于篇幅的原因，具体的安装教程可以参考大佬的文章，不过要注意一点！由于我们要在Vscode上使用Git的，所以在安装Editor时要选择Vscode。



安装结束后在任意目录下右键出现如下图所示的两个菜单，那么就说明安装成功了。



返回顶部

## 使用Git

## 1.配置基本信息

选择一个文件夹（为了方便描述我将其命名为`localRepository`），打开 *Git Bash*，当使用Git时首先要做的事情是设置用户名称和email地址，因为每次Git提交都会使用该用户信息。

输入

```
1 ##设置用户名:
2 git config --global user.name 'XXX'
3
4 ##设置邮箱:
5 git config --global user.email 'XXX'
```

如果没有出现任何反应就是成功了。

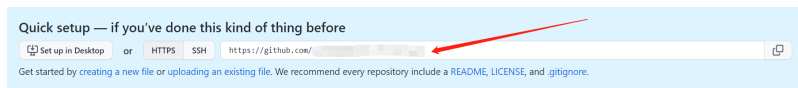
## 2.连接GitHub仓库

在一开始我是分不清楚Git、GitHub甚至于Gitee之间的关系的，相信有初学的小伙伴也是如此，那么我先简单介绍什么是Git，而什么又是GitHub。

Git是分布式版本控制系统，GitHub是一个基于Git的代码托管平台。也就是说我可以通过Git来管理存在的多种版本的代码（并且由于各种原因这些版本都是想要保留的），而且能上传至GitHub（或者类似的平台），作为保管或者供以社区分享。

1、登录GitHub（或者是Gitee之类的平台），点击头像——>选择 *Your repositories*，新建仓库。

这里我们要复制仓库的地址，之后连接仓库。



2、输入

```
git clone 'XXX'(仓库地址)
```

**clone**是将远程仓库连接到本地的指令。

如果出现当`time out443`或者`unable to access 'XXX': OpenSSL SSL_read: Connection was aborted, errno 10053`时，尝试输入以下指令：

```
1 git config --global --unset http.proxy
2 git config --global --unset https.proxy
3 ##当然这样治标不治本，要完全的解决这种问题，需要科学上网
```

如果连接的这个仓库是新建的，会出现如下Warning，不用担心这是因为我们新建的仓库中并没有存放东西。

```
warning: You appear to have cloned an empty repository.
```

除此之外没有其他的报错或者警告就说明我们clone成功了。

## 3.添加、查看和提交

这时的`localRepository`文件夹中会出现一个与你远程仓库同名的文件夹，现在我们要在Bash中打开这个文件夹。

直到路径后面出现 **(main | master)**

```
1 cd XXX（文件所在的路径）
2
##tips: 注意在Bash中，用“/”正斜杠分层；使用.表示当前文件的目录，而..表示当前文
```

返回顶部

此时我在文件夹中**新建**一个名为“README”的文本文件，**使用add将选定的文件添加至缓存区。**

```
1 | git add README.txt
2 | ##也可以使用“git add .”添加当前目录下所有未提交的文件
```

我们可以使用status查看当前仓库的状态。这时我们发现README.txt的字样变成**绿色**了，而在添加之前则是**红色**的。

```
git status
```

```
$ git status
On branch main

No commits yet

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

可以看见它提示我们该文件虽然添加了，但还不算是添加到了本地仓库，我们还需要使用git commit命令为其添加修改的描述信息。

```
1 | git commit -m 'add README.txt'
2 | git commit README.txt -m 'add README.txt'
3 | ##二者皆可，后者说明具体提交哪个文件，但是commit会提交所有修改的文件
```

**注意-m'XXX'一定要有，表示提交描述信息。**

```
$ git commit -m 'add README'
[main (root-commit) f4676d4] add README
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 README.txt
```

出现**1 file changed**，表示我们提交成功了。

#### 4.查看历史版本和滚回

这时我在文本中写点东西，再次提交并且描述信息是“update”。如果我们想要查看该文件的历史提交版本和提交信息，需要用log指令。

```
git log XXX
```

```
git log
commit 28af0007338ad4dff88f38597aeecc5eccfb372d3 (HEAD -> main)
Author: 00000001-1571-154055@qq.com
Date: Tue Nov 1 15:23:39 2022 +0800

    update

commit f4676d4c6619e82bc187f1bb4512487d77f7b332
Author: 00000001-1571-154055@qq.com
Date: Tue Nov 1 15:12:55 2022 +0800

    add README
```

可以看见**第一次提交**的描述“**add README**”和**第二次提交**的描述“**update**”，以及两次提交的时间和哈希值。

然后按下‘q’键退出菜单。

这里再说明一下图中的HEAD和main的含义：

main表示仓库中的主分支，HEAD可以抽象成一个活动指针，指向当前所活动的分支。

[返回顶部](#)

那我们想要将文件回滚到之前的某个版本该怎么办呢？记录要回滚版本的哈希值（可以只是前几位数），使用checkout将HEAD指针回到该历史版本。

```
1 ##将HEAD指针指向想要回滚的历史版本
2 git checkout XXX（哈希值）
3 ##将版本回滚至HEAD指向的版本
4 git reset HEAD
```

回滚不代表Git会删除本地的历史记录，所以此时只要reset to main还是可以回到最新的版本。

而使用`reset --hard`指令可以重置回到某个历史版本，注意使用这个指令前请确保当前的版本已经提交（commit）了，原因后面会说。

```
git reset --hard XXX（哈希值）
```

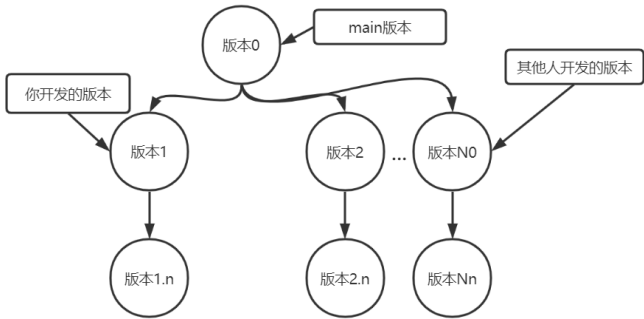
如果使用`--hard`，那么一切就全都恢复了，HEAD指针变了，add的暂存区消失，代码什么的也恢复到以前状态。

我们可以通过`reflog`指令查看所有的提交过的历史版本，其中包括我们回滚的版本。若是想要“撤销”刚才的回滚操作，我们可以试试这个指令。

```
git reflog
```

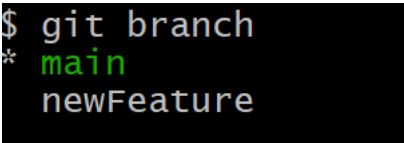
### 5. 分支

实际上当我们在团队中或者公司中开发项目时，不可能总是一条直线型的版本开发，更多的是通过在main分支上建立新的分支，然后在这个新的分支中开发自己工作内的代码，最后再将二者合并。



使用branch指令查看当前的分支，使用branch name指令建立新的分支。

```
1 ##查看所有分支
2 git branch
3 ##新建分支
4 git branch XXX（新分支的名字）
```



\*main表示这是当前我们所在的分支，而newFeature是我创建的新的分支。

我们同样可以通过checkout检出到newFeature分支上，然后在文本写下“编辑者：小白”，再添加提交，得到下图①：

返回顶部

```
commit 7688330f751de86147bd25c61cae763310b6429f (HEAD -> newFeature)
Author: 30000001 15711510556
Date: Tue Nov 1 17:02:00 2022 +0800

    write by xiaobai

commit b354f4770ef22c5fd8dafca05aad2997da543322 (origin/main, main)
Author: 30000001 15711510556
Date: Tue Nov 1 15:45:20 2022 +0800

    update

commit f4676d4c6619e82bc187f1bb4512487d77f7b332
Author: 30000001 15711510556
Date: Tue Nov 1 15:12:55 2022 +0800
```

(这里请忽视掉origin/main的字符，我稍后会提到)

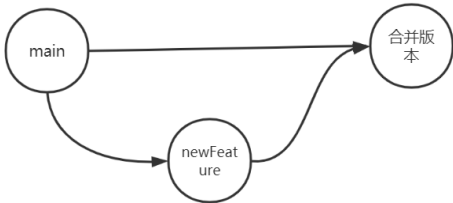
而如果这时我们又想要删除掉这个分支，使用branch -D的指令。

```
git branch -D XXX (分支的名字)
```

这时在分支列表中的newFeature已经删除了，并且文本也回到了建立分支前的状态。

### 6.合并

现在回到①，这时我们想要将newFeature分支和main的分支合并，那么要使用merge的指令。但是这里要先注意，我们是想要在main分支上看到我们在newFeature分支上作出的修改，那么我们要先指回main分支，再将newFeature分支合并。



```
1 ##指向main
2 git checkout main
3 ##合并
4 git merge newFeature
```

### 7.上传和拉取

当我们完成在本地仓库的操作之后，就需要上传至远程仓库，那这时我们需要用到push指令。

```
git push
```

若是出现这样像是“乱码”一样的东西就是上传成功了，这时我们回到GitHub上的仓库，会发现刚刚上传的文件。

```
$ git push
Enumerating objects: 175, done.
Counting objects: 100% (175/175), done.
Delta compression using up to 16 threads
Compressing objects: 100% (171/171), done.
Writing objects: 100% (174/174), 44.54 MiB | 932.00 KiB/s, done.
Total 174 (delta 12), reused 1 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (12/12), done.
To 30000001 15711510556
main -> main
```

tips: 注意啦，如果项目是用VS写的，那么建议先删除VS生成的临时文件以减小文件大小，这样子更容易上传成功。

返回顶部

那当我们的同伴修改了我们上传的代码，而我们想要在他的版本上继续修改时，需要用到pull指令，pull是将远程仓库分支中修改的内容下拉并合并到本地仓库中。

```
git pull
```

8.总结

到这为止，我们已经学习了80%实际开发中的需要掌握的Git操作，剩下的20%更多是处理复杂的工作流程的能力(或者当你陷入突发情况时)，当然这已经不是本篇文章的内容了。

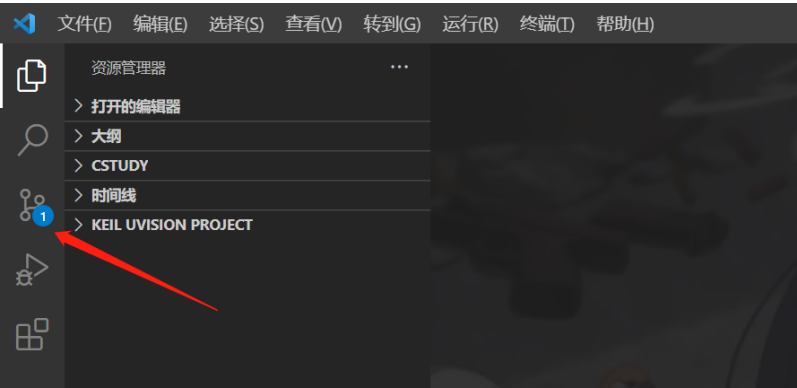
为此我特意总结了如下资源，供以那些想要系统学习Git的小伙伴。

- 免费的教程网站：
  - Git官方文档
  - Git简明指南
  - 可视化的Git操作学习
- 站内大佬文章：
  - 关于Git这一篇就够了

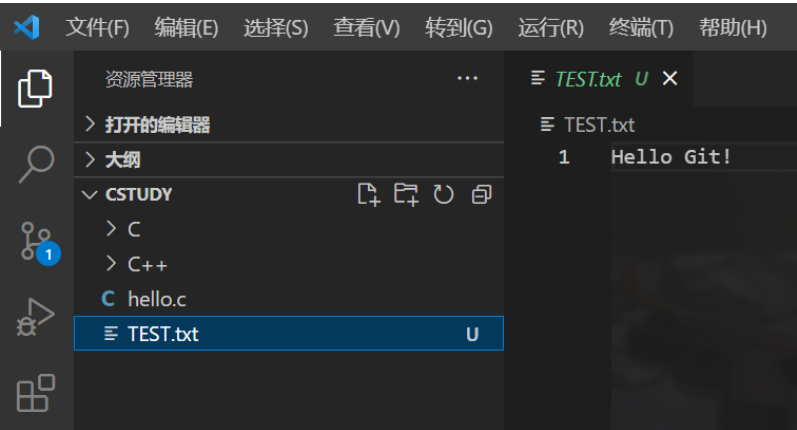
Vscode使用Git

使用Vscode+Git可以让我们更方便的使用Git，也可以更直观的将代码的变化（比较）展示出来。

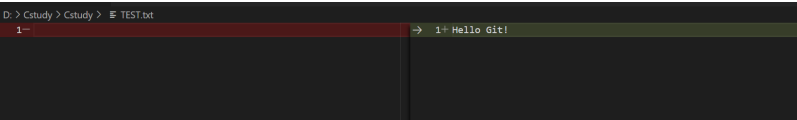
如果你按照正确的步骤安装完Git之后，会发现Vscode中【源代码管理】一栏变得可用了（位于最左边菜单栏中的第三个），这就是我们在VScode中使用Git的场景。



当我用Vscode打开在上文中提到的文本文件时，会发现文件名后面跟了一个字母‘U’，这表示该文件尚未追踪，说白了，就是在Bash中没有使用add指令添加的意思。

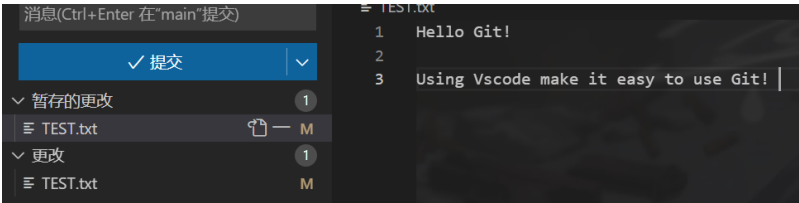


我们打开【源代码管理】一栏，点击文件名后面的‘+’号，发现字母U变成了‘A’，表示已经成功add了。这时Vscode会把文件修改的部分直观的展示在我们面前，十分方便。

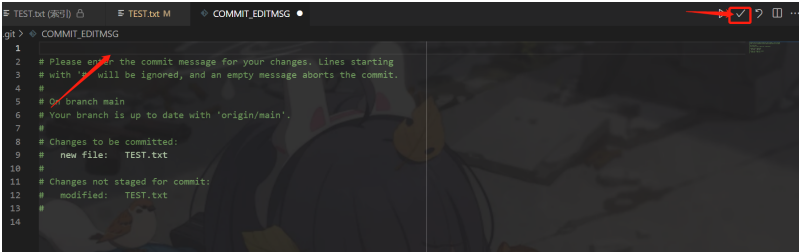


而当我再一次修改文本时，会发现字母A又变成了字母‘M’，表示我修改了它。

返回顶部



我们再次点击加号添加，然后点击“提交”，表示我进行了commit。那么既然是commit就一定要有提交信息是吧，我们在新弹出的窗口中的最上边一行写入本次提交的信息，然后在上面的菜单栏的右边，点击一个“✓”的按钮，保存更改，表示提交成功了。



这时我们发现菜单栏的高亮消失了。最后我们点击左侧菜单上方的“...”，会发现我们上文学习的指令都在这里打包好了，最后我们点击“推送”，回到GitHub中发现该文本已经顺利推送了。

Vscode已经将指令封装成按钮了，其他的操作流程，和在Bash中是一样的（所以Bash还是要学滴！），我们同样可以在终端中输入指令实现操作。

文章知识点与官方知识档案匹配，可进一步学习相关知识

Git技能树 Git入门 Git简介 5819 人正在系统学习中

显示推荐内容

觉得还不错? 一键收藏

 **小白还在写代码** 关注 13 73 2 专栏目录

**Vscode配置git\_渡口五十六的博客** 8-11

**Vscode如何配置Git AshleyXM的博客** 6078 第一步:配置Git环境变量右键“我的电脑”,选择“属性”...

一文吃透VSCode中git的相关配置与使用\_vscode git\_qq\_43067585的博客-C... 8-10  
详细介绍了GIT的使用方式,包括GIT介绍、安装、相关的命令解释,还有项目上传至云端与克隆...

**vscode集成Git操作 最新发布** qq\_57200471的博客 394  
4.点击“✓”号,右边的箭头输入需要备注的信息。然后按 Enter 确定;1.将远程仓库 git clone 仓库...

**Windows+VScode配置与使用git, 超详细教程, 赶...** 热门推荐 czjl6886的博客 5万+  
目录 第一步: 安装Git命令行工具 第二步: 配置VScode中的git 第三步: 使用 VScode + git, 提...

**vscode中git的配置\_gittab wescode\_兰亭古墨的博客** 7-30  
1.vscode中git的配置 首先需要你的电脑上已经安装了git,且window电脑里git添加到环境变量中...

**vscode配置、使用git\_vscode .git/config\_aら 淼的博客** 8-5  
git config --global user.email “你的邮箱” 二、vscode配置并使用git 1、打开vscode,按下快捷键 ...

**vsCode中Git的使用** qianqianyunyan的博客 2058  
vsCode中Git的使用心得

**手把手教你在VSCode中使用Git** TYINY的博客 3万+  
我本身也多次看到他们用vscode查看修改prometheus代码 摘自: https://mp.weixin.qq.com/s/D...

**VScode如何配置Git** AshleyXM的博客 6079  
第一步: 配置Git环境变量 右键“我的电脑”, 选择“属性”, 在弹出的对话框左侧点击“高级系统设...

**VSCode配置Git (转载)** 圆圆的 JavaScript 博客 1629  
在vscode中使用git, 首先要把vscode的git配置做好

**VScode中配置git** qq\_40948117的博客 1553

返回顶部

一、VS code 配置git 1. 下载安装git 2. 如果要在VS Code里面使用Git则需要在编辑器内配置git....

在VSCode使用git

chen110710的博客 7452

第五步：写入git@gitee.com:XXXXXXX/gittest.git。 ， 这是Git默认的叫法， 也可以改成别的， ...

vscode+git+node

vscode+git+node

11-24

基于vscode的git pull后代码冲突的两种解决方式

08-21

在vscode上针对git可视化操作,在git pull后出现冲突的解决方式

git-commit-lint-vscode:vscode一款git 规范化提交插件

05-17

git-commit-lint-vscode 在日常的开发中,目前主流的代码管理工具就是 git 了,当我们对代码进行...

vscode环境中配置git

allen1880的专栏 6257

1. 建立project空目录用于存放electron基本环境这个环境中得文件不能加入git项目中否则每次...

vscode 配置git

weixin\_43507141的博客 4084

下载git https://git-scm.com/ 安装时， 直接默认所有选项安装 然后打开git安装目录 找到如下路...

vscode配置git

浪子天涯梦的博客 8808

首先下载git 下载完成后复制git路径， 如： D:\\Software\\Git\\cmd\\git.exe 也可以打开cmd窗口...