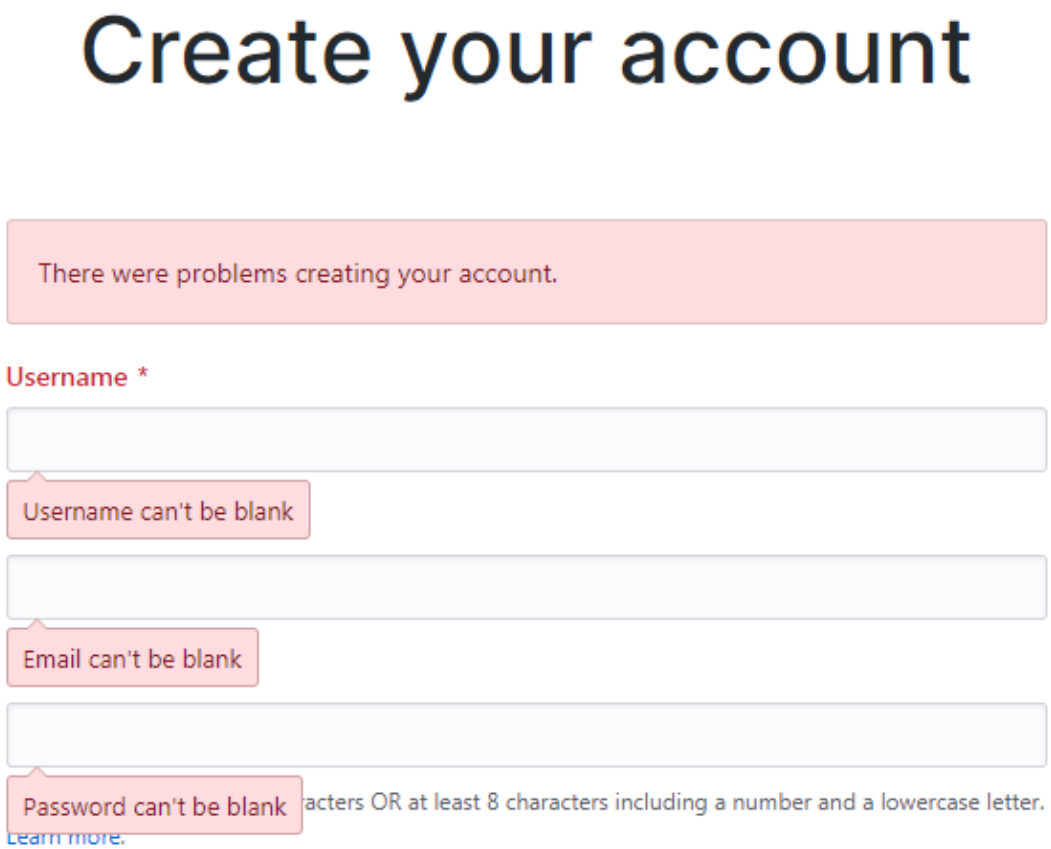
github

```
内容：
1).github注册账号
2).安装git服务端，客户端
3).命令操作上传文件至开源平台github
4).使用git客户与账号，查看（下载）文件
```

github注册账号

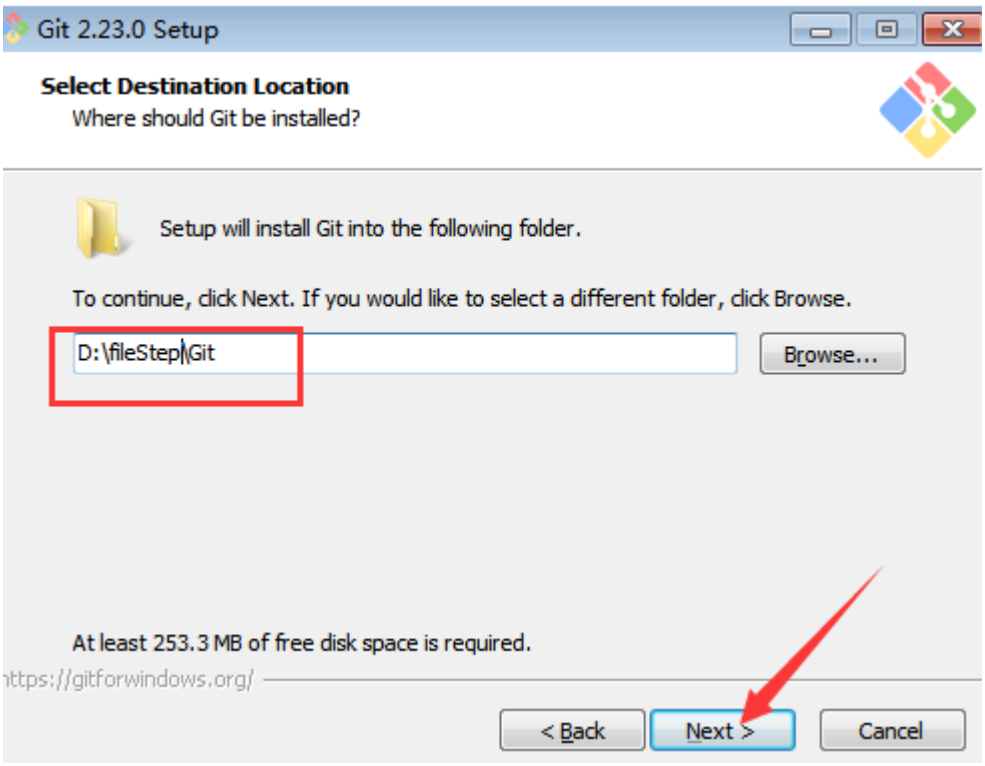
浏览器打开 https://github.com/join 注册一个账号，后期安装git客户端时直接输入记录。

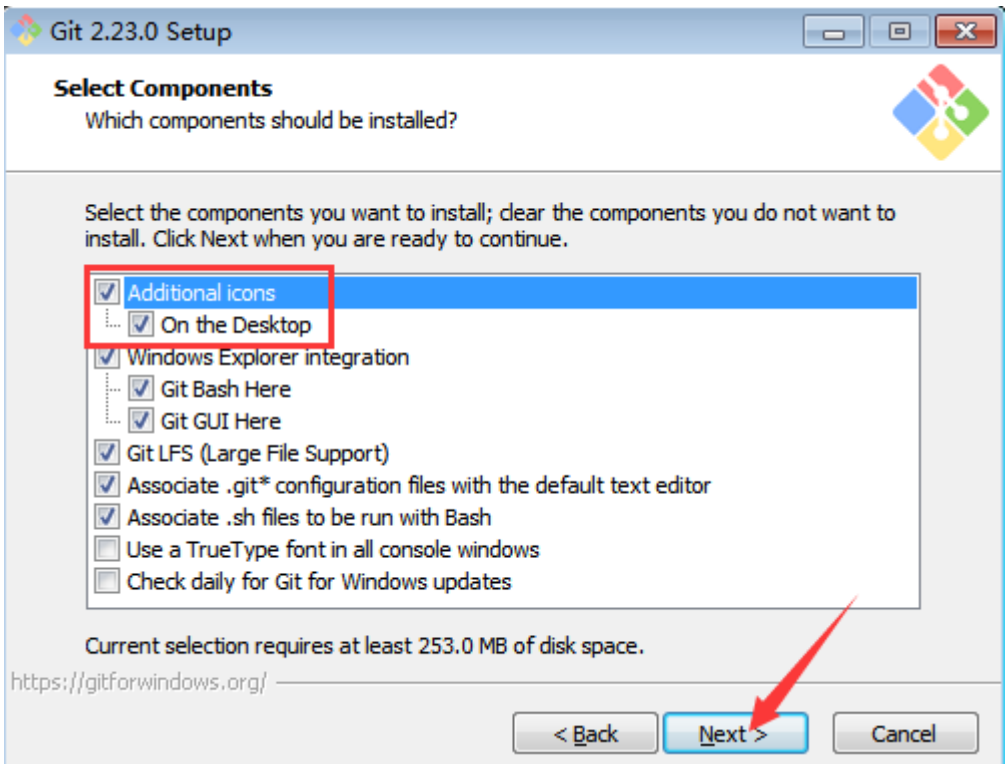


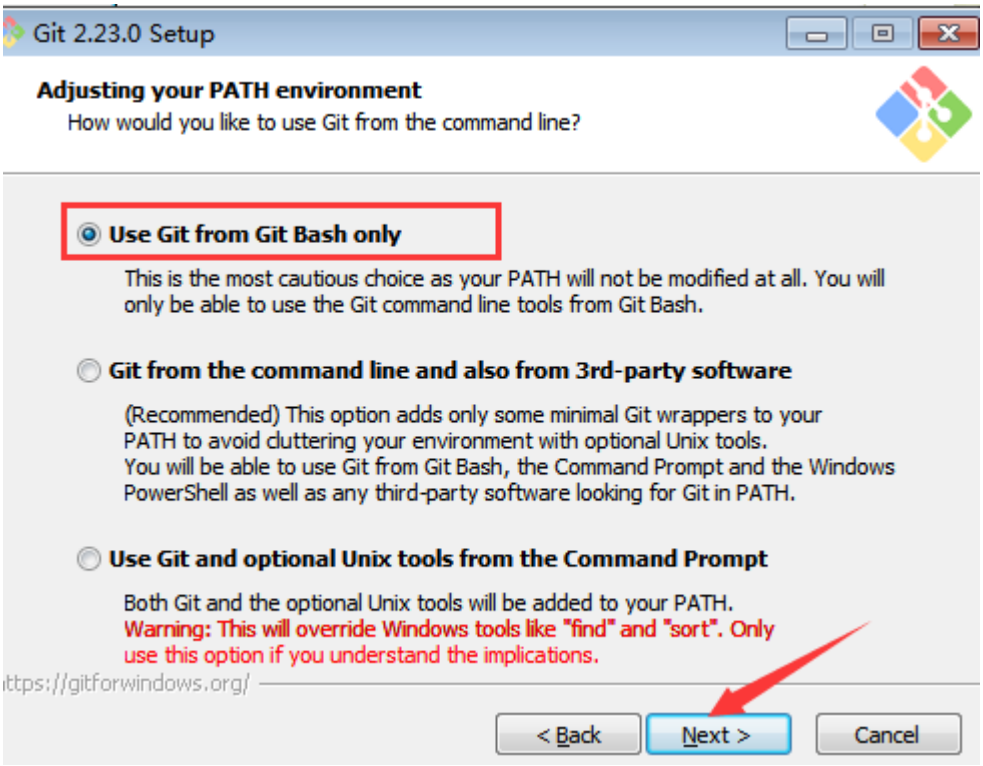
打开邮箱地址进行激活github账号方可有效：

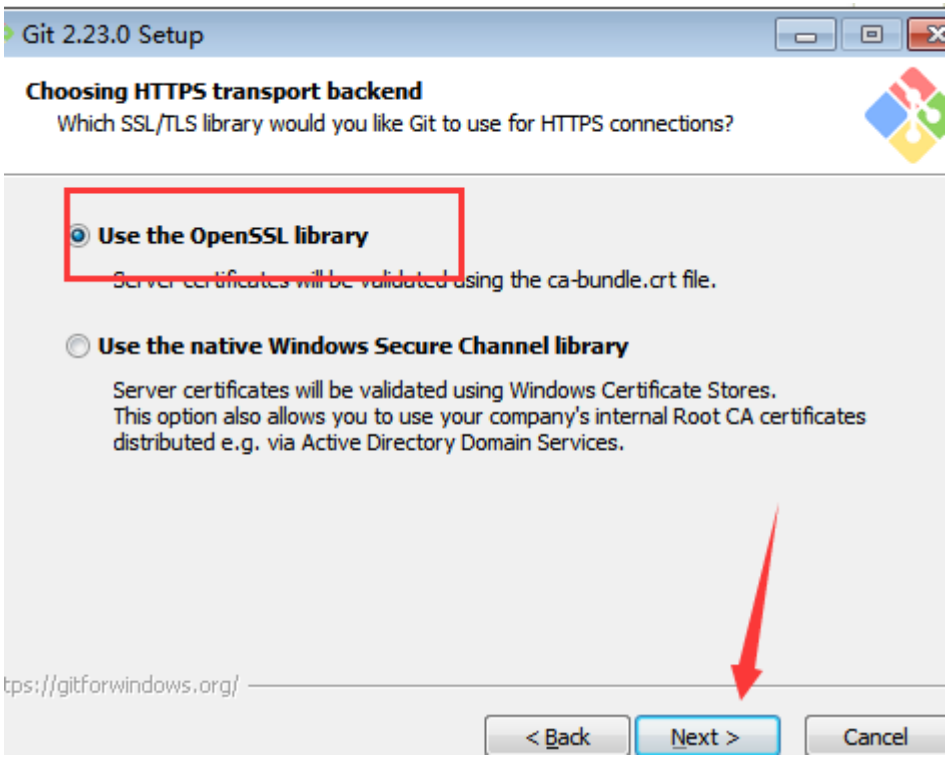
安装git服务端

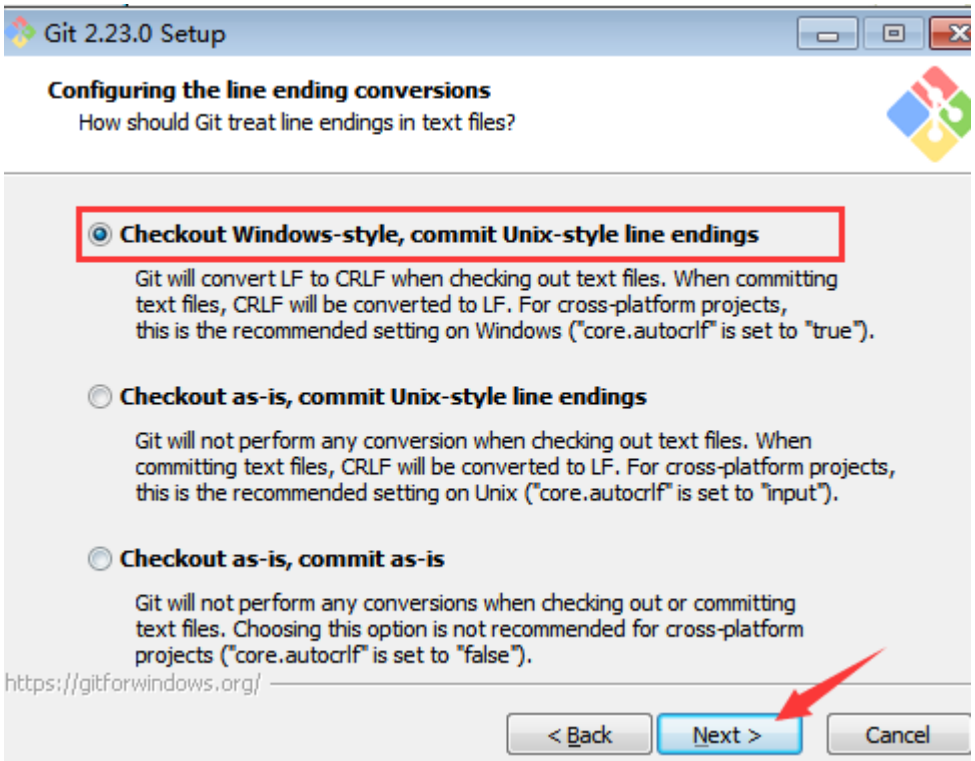
找到**Git-2.23.0-64-bit.exe**点击以管理身份运行安装：

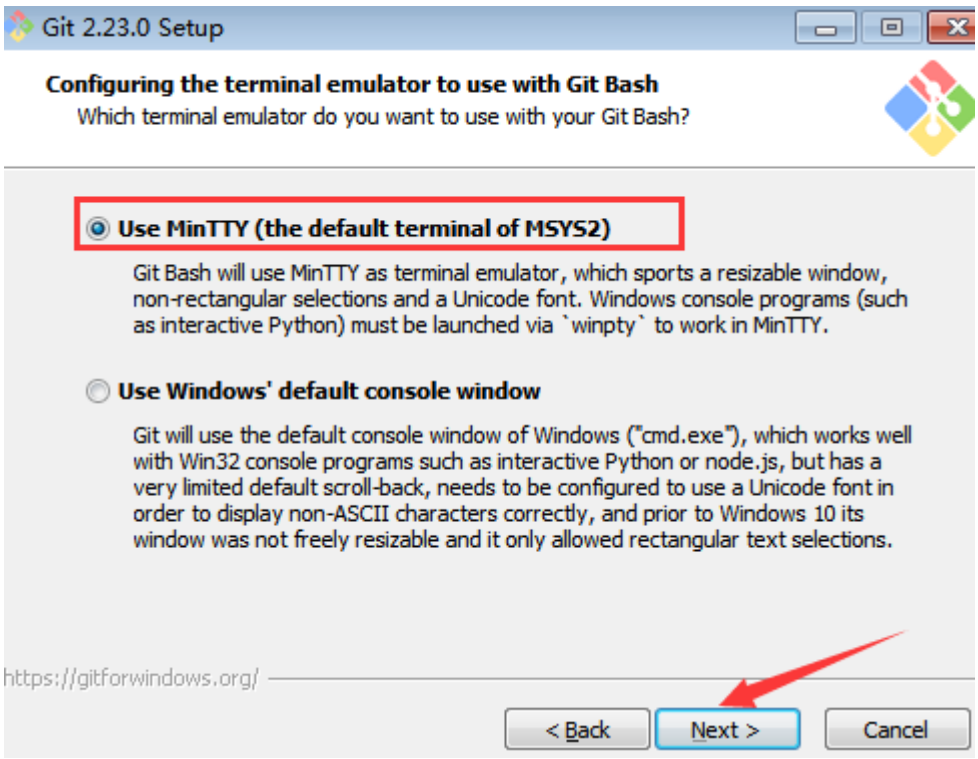

Git 2.23.0 Setup
Select Destination Location
Where should Git be installed?
Setup will install Git into the following folder.
To continue, click Next. If you would like to select a different folder, click Browse.
D:\fileStep\Git
Browse...
At least 253.3 MB of free disk space is required.
https://gitforwindows.org/
< Back
Next >
Cancel

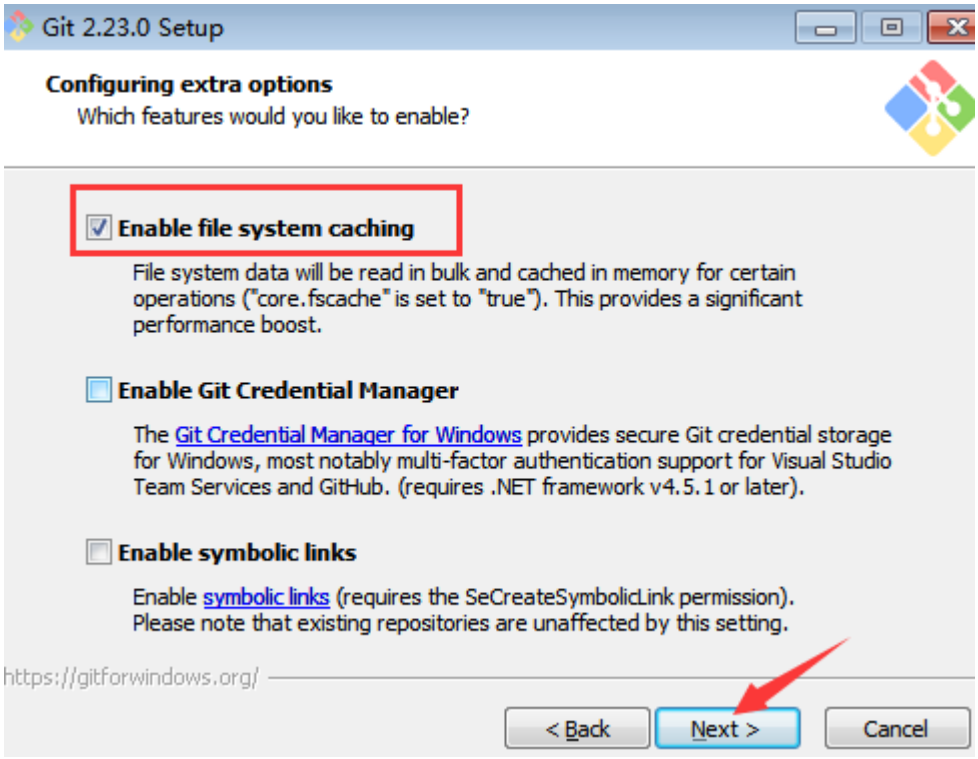

Git 2.23.0 Setup
Select Components
Which components should be installed?
Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.
Additional icons
On the Desktop
Windows Explorer integration
Git Bash Here
Git GUI Here
Git LFS (Large File Support)
Associate .git* configuration files with the default text editor
Associate .sh files to be run with Bash
Use a TrueType font in all console windows
Check daily for Git for Windows updates
Current selection requires at least 253.0 MB of disk space.
https://gitforwindows.org/
< Back
Next >
Cancel

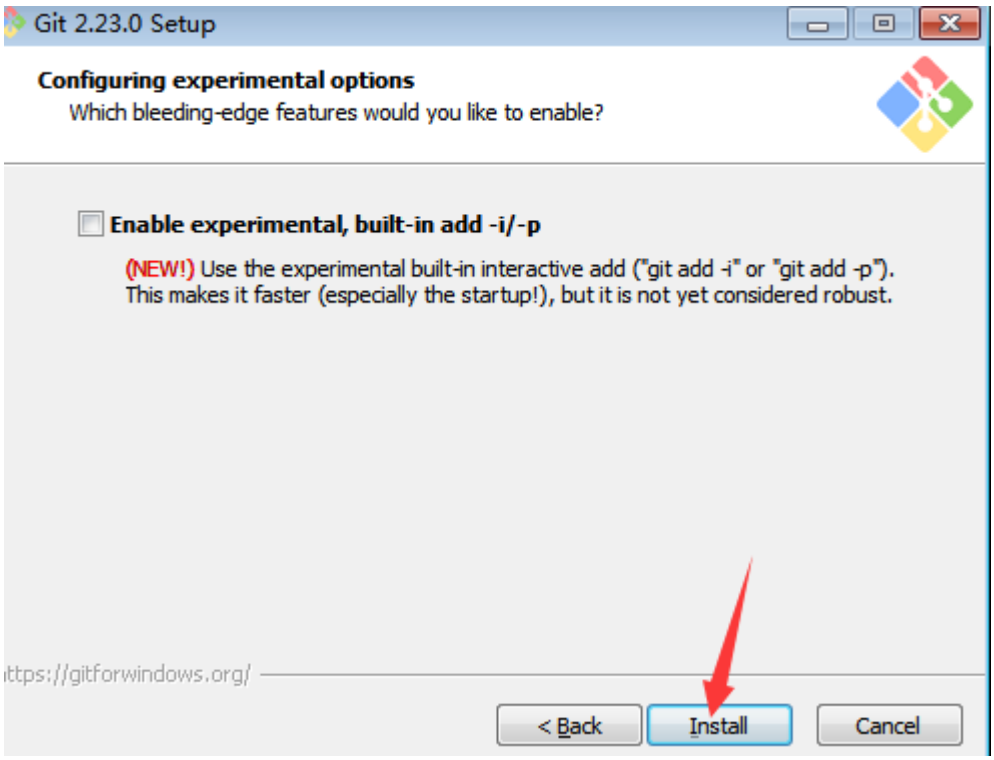

Git 2.23.0 Setup
Adjusting your PATH environment
How would you like to use Git from the command line?
Use Git from Git Bash only
This is the most cautious choice as your PATH will not be modified at all. You will only be able to use the Git command line tools from Git Bash.
Git from the command line and also from 3rd-party software
(Recommended) This option adds only some minimal Git wrappers to your PATH to avoid cluttering your environment with optional Unix tools.
You will be able to use Git from Git Bash, the Command Prompt and the Windows PowerShell as well as any third-party software looking for Git in PATH.
Use Git and optional Unix tools from the Command Prompt
Both Git and the optional Unix tools will be added to your PATH.
Warning: This will override Windows tools like "find" and "sort". Only use this option if you understand the implications.
ttps://gitforwindows.org/
< Back
Next >
Cancel

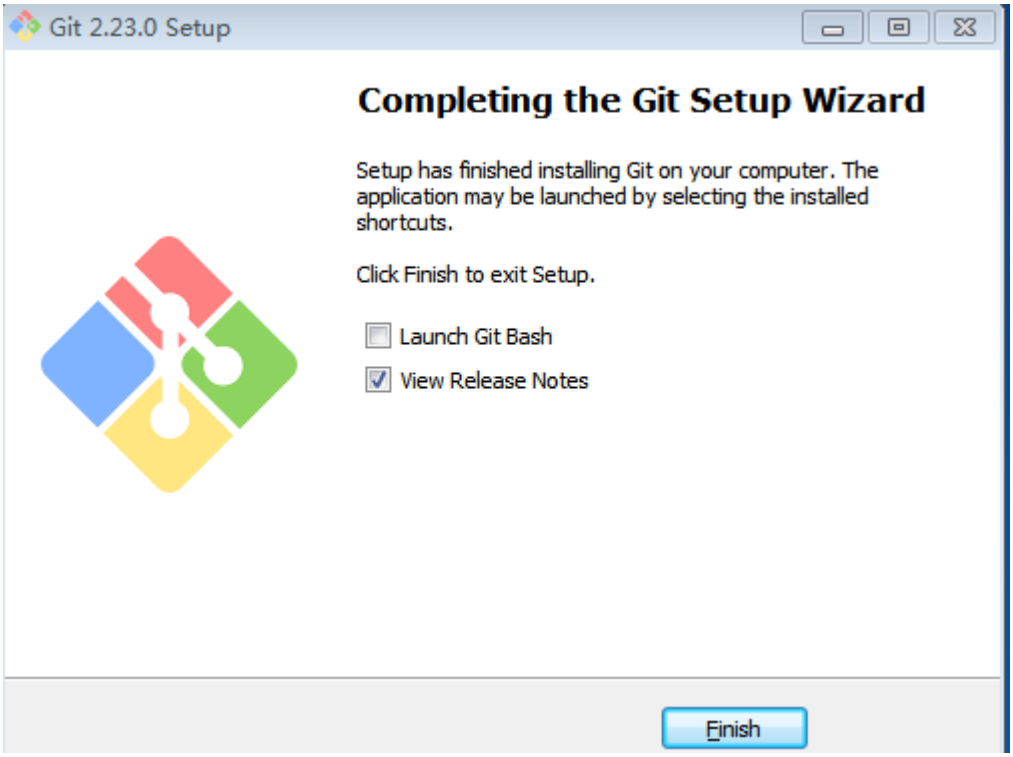

Git 2.23.0 Setup
Choosing HTTPS transport backend
Which SSL/TLS library would you like Git to use for HTTPS connections?
Use the OpenSSL library
Server certificates will be validated using the ca-bundle.crt file.
Use the native Windows Secure Channel library
Server certificates will be validated using Windows Certificate Stores.
This option also allows you to use your company's internal Root CA certificates distributed e.g. via Active Directory Domain Services.
tps://gitforwindows.org/
< Back
Next >
Cancel


Git 2.23.0 Setup
Configuring the line ending conversions
How should Git treat line endings in text files?
Checkout Windows-style, commit Unix-style line endings
Git will convert LF to CRLF when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Windows ("core.autocrlf" is set to "true").
Checkout as-is, commit Unix-style line endings
Git will not perform any conversion when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Unix ("core.autocrlf" is set to "input").
Checkout as-is, commit as-is
Git will not perform any conversions when checking out or committing text files. Choosing this option is not recommended for cross-platform projects ("core.autocrlf" is set to "false").
https://gitforwindows.org/
< Back
Next >
Cancel


Git 2.23.0 Setup
Configuring the terminal emulator to use with Git Bash
Which terminal emulator do you want to use with your Git Bash?
Use MinTTY (the default terminal of MSYS2)
Git Bash will use MinTTY as terminal emulator, which sports a resizable window, non-rectangular selections and a Unicode font. Windows console programs (such as interactive Python) must be launched via `winpty` to work in MinTTY.
Use Windows' default console window
Git will use the default console window of Windows ("cmd.exe"), which works well with Win32 console programs such as interactive Python or node.js, but has a very limited default scroll-back, needs to be configured to use a Unicode font in order to display non-ASCII characters correctly, and prior to Windows 10 its window was not freely resizable and it only allowed rectangular text selections.
https://gitforwindows.org/
< Back
Next >
Cancel


Git 2.23.0 Setup
Configuring extra options
Which features would you like to enable?
Enable file system caching
File system data will be read in bulk and cached in memory for certain operations ("core.fscache" is set to "true"). This provides a significant performance boost.
Enable Git Credential Manager
The Git Credential Manager for Windows provides secure Git credential storage for Windows, most notably multi-factor authentication support for Visual Studio Team Services and GitHub. (requires .NET framework v4.5.1 or later).
Enable symbolic links
Enable symbolic links (requires the SeCreateSymbolicLink permission). Please note that existing repositories are unaffected by this setting.
https://gitforwindows.org/
< Back
Next >
Cancel

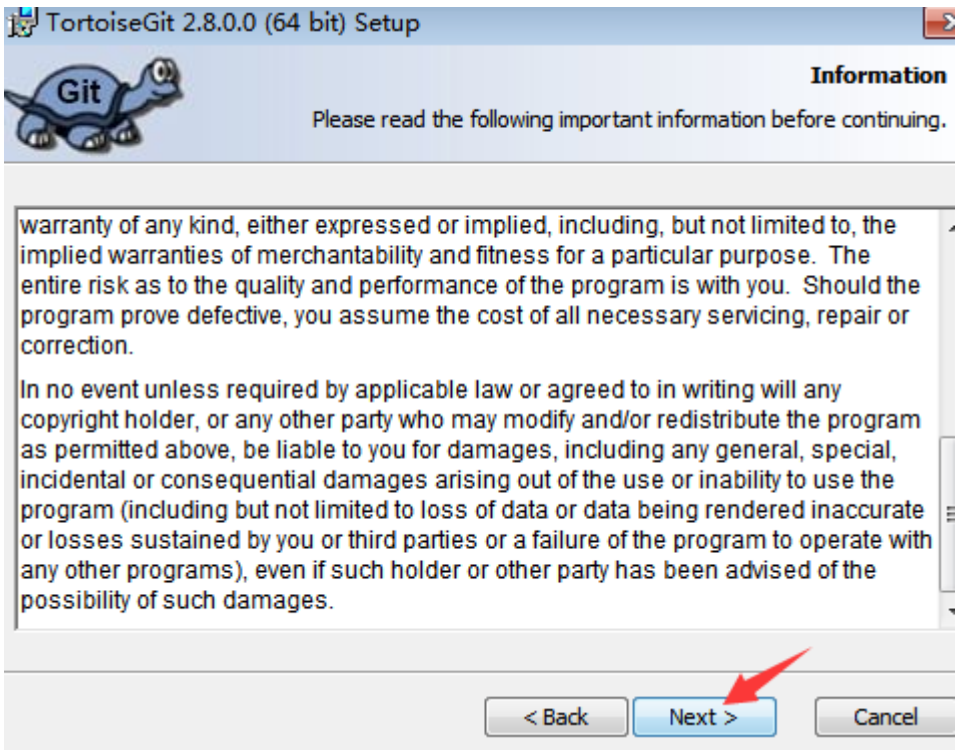

Git 2.23.0 Setup
Configuring experimental options
Which bleeding-edge features would you like to enable?
Enable experimental, built-in add -i/-p
(NEW!) Use the experimental built-in interactive add ("git add -i" or "git add -p"). This makes it faster (especially the startup!), but it is not yet considered robust.
https://gitforwindows.org/
< Back
Install
Cancel

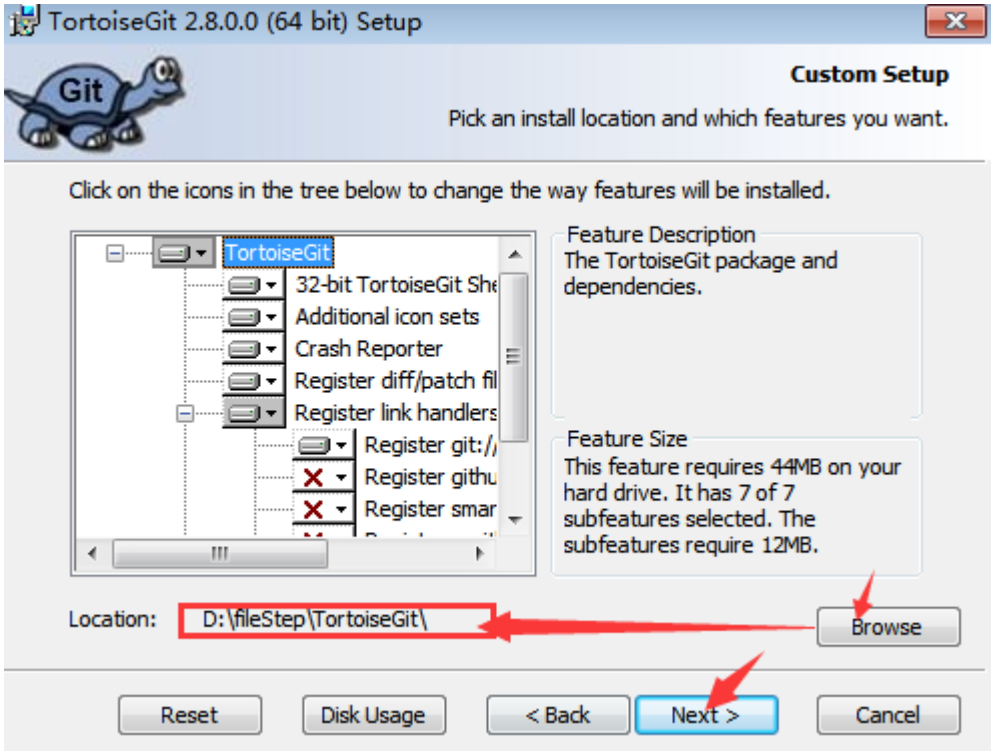


安装git客户端

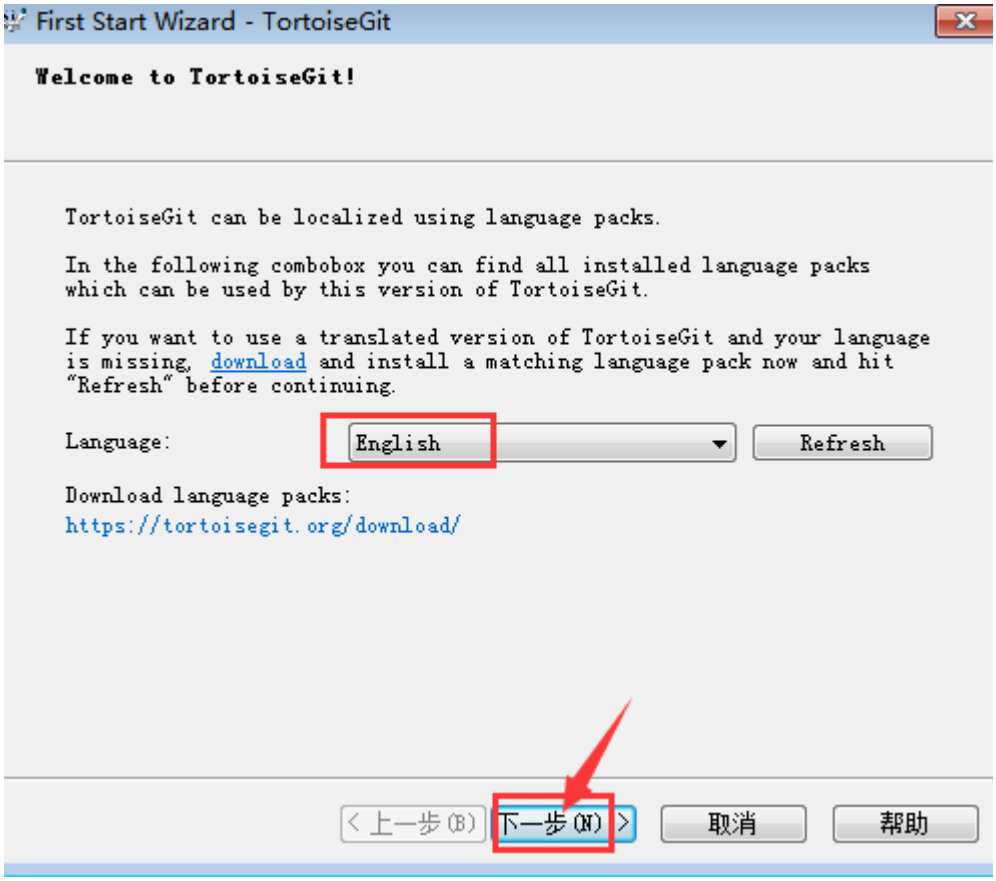
找到**TortoiseGit-2.8.0.0-64bit.msi**安装包，点击以身份管理员安装：

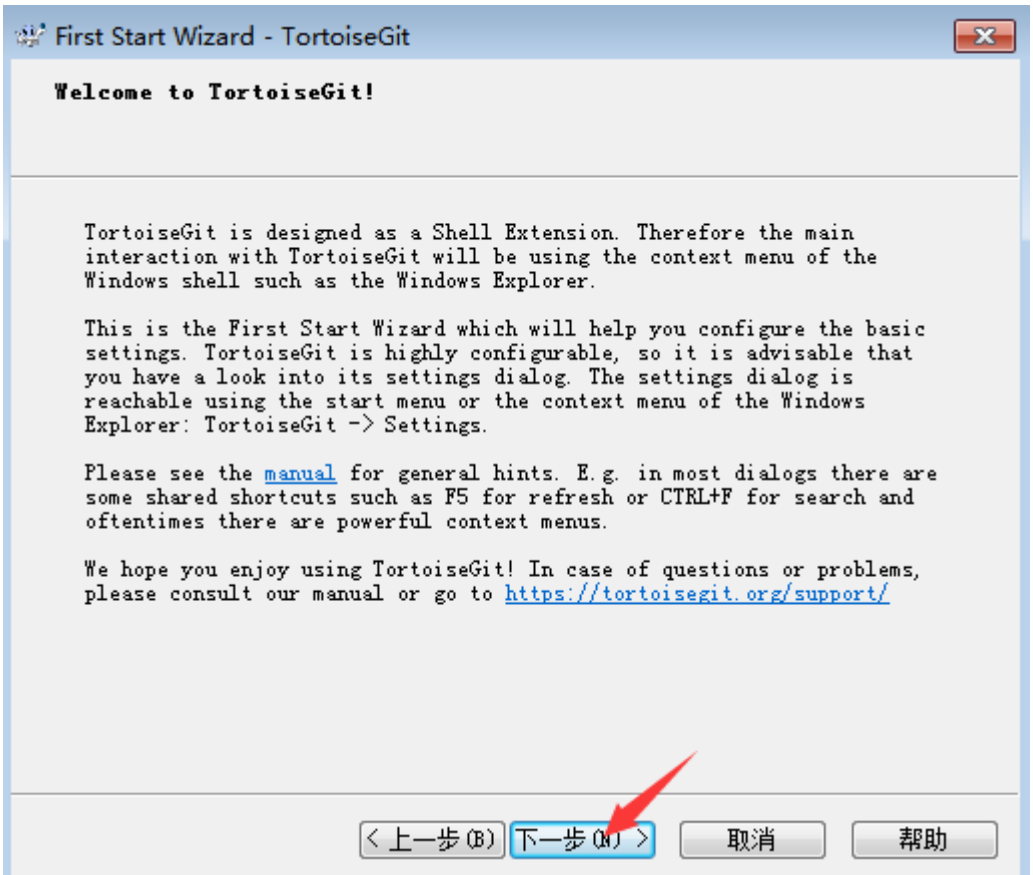

TortoiseGit 2.8.0.0 (64 bit) Setup
Git
Welcome to the TortoiseGit 2.8.0.0 (64 bit) Setup Wizard
The Setup Wizard will install TortoiseGit 2.8.0.0 (64 bit) on your computer. Click Next to continue or Cancel to exit the Setup Wizard.
< Back
Next >
Cancel

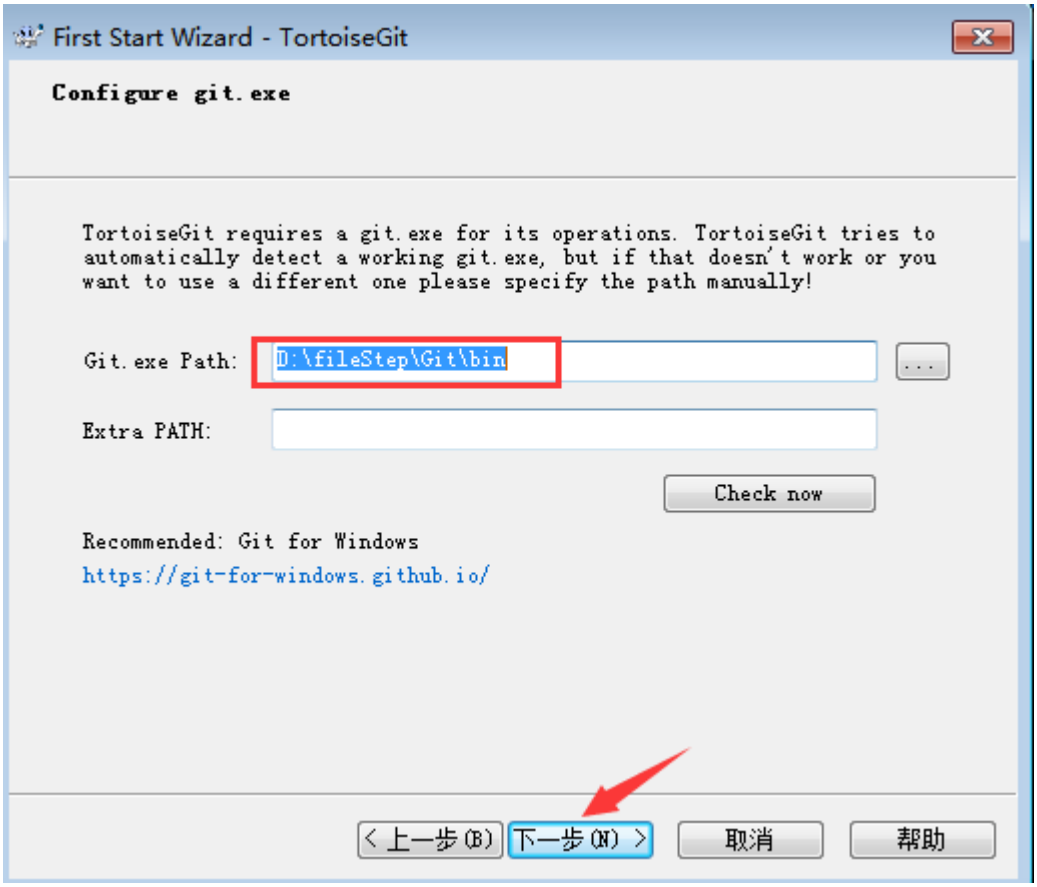

TortoiseGit 2.8.0.0 (64 bit) Setup
Git
Information
Please read the following important information before continuing.
warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of the program is with you. Should the program prove defective, you assume the cost of all necessary servicing, repair or correction.
In no event unless required by applicable law or agreed to in writing will any copyright holder, or any other party who may modify and/or redistribute the program as permitted above, be liable to you for damages, including any general, special, incidental or consequential damages arising out of the use or inability to use the program (including but not limited to loss of data or data being rendered inaccurate or losses sustained by you or third parties or a failure of the program to operate with any other programs), even if such holder or other party has been advised of the possibility of such damages.
< Back
Next >
Cancel


TortoiseGit 2.8.0.0 (64 bit) Setup
Custom Setup
Pick an install location and which features you want.
Click on the icons in the tree below to change the way features will be installed.
TortoiseGit
32-bit TortoiseGit She
Additional icon sets
Crash Reporter
Register diff/patch fil
Register link handlers
Register git:/
Register githu
Register smar
Feature Description
The TortoiseGit package and dependencies.
Feature Size
This feature requires 44MB on your hard drive. It has 7 of 7 subfeatures selected. The subfeatures require 12MB.
Location:
D:\fileStep\TortoiseGit\
Browse
Reset
Disk Usage
< Back
Next >
Cancel

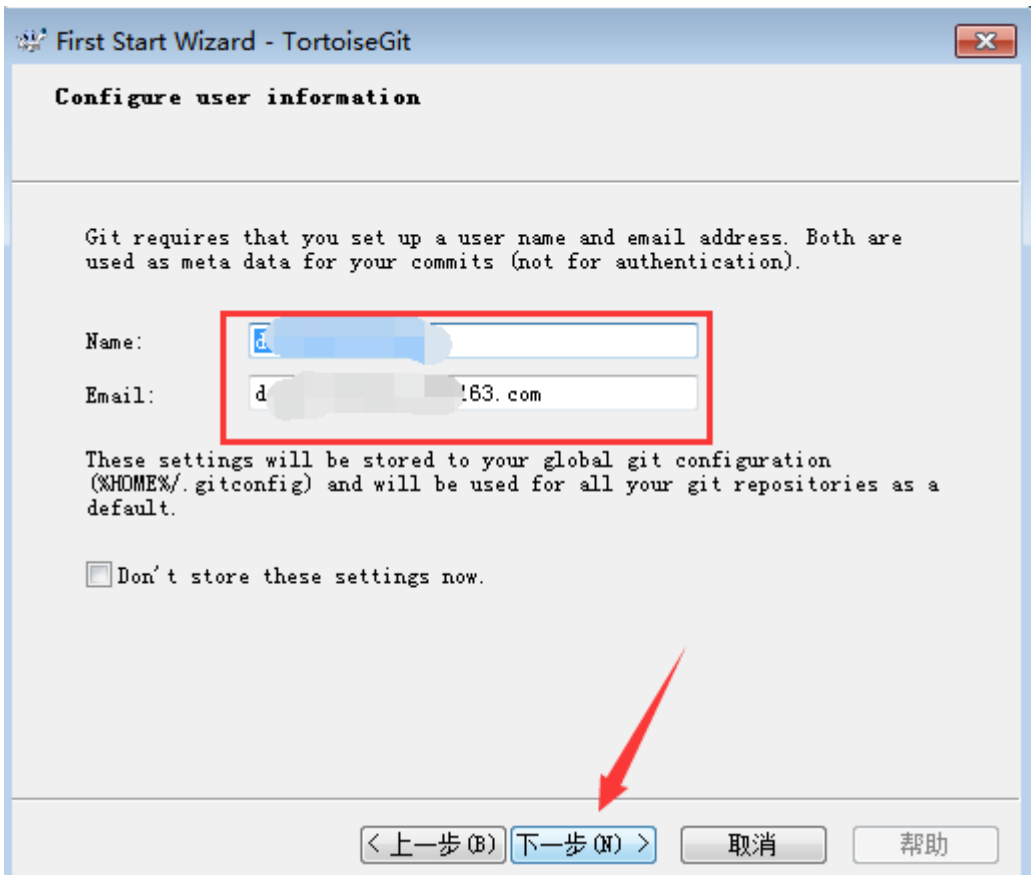

TortoiseGit 2.8.0.0 (64 bit) Setup
Completing the TortoiseGit 2.8.0.0 (64 bit) Setup Wizard
Click the Finish button to exit the Setup Wizard.
Show Changelog
Run first start wizard
Thanks for using TortoiseGit. You can show your appreciation and support future development by donating!
Donate!
< Back
Finish
Cancel

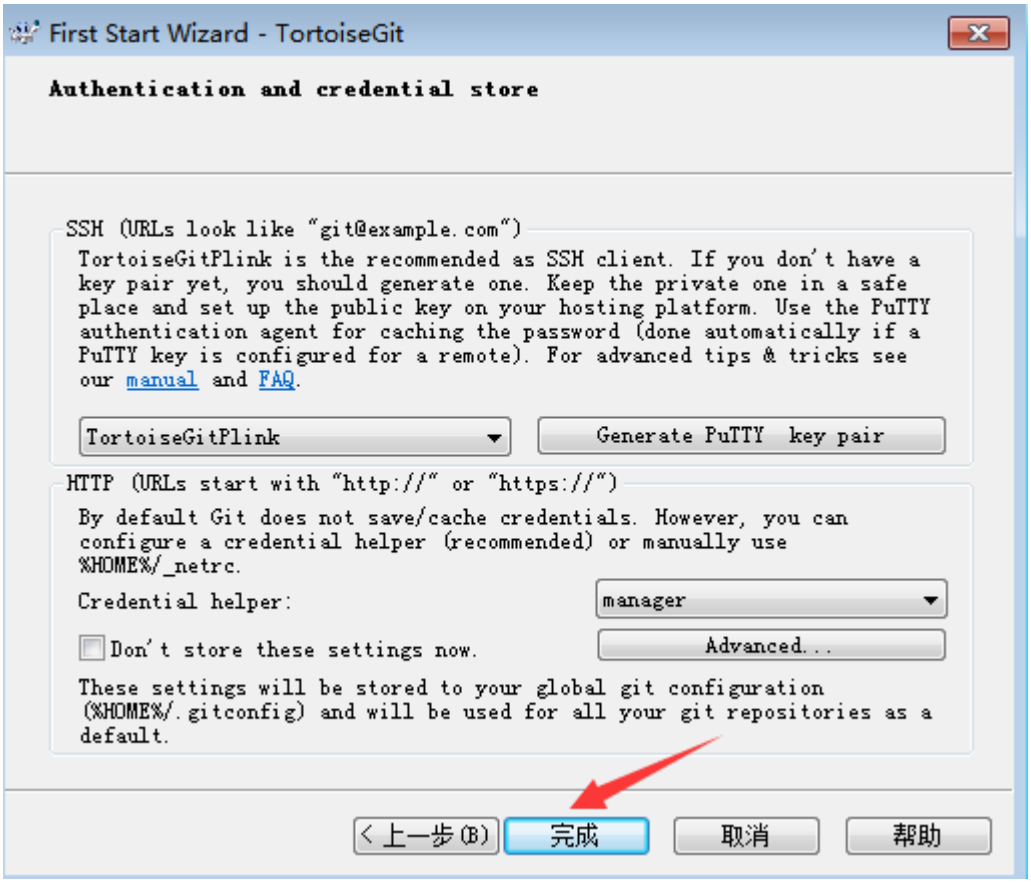

First Start Wizard - TortoiseGit
Welcome to TortoiseGit!
TortoiseGit can be localized using language packs.
In the following combobox you can find all installed language packs which can be used by this version of TortoiseGit.
If you want to use a translated version of TortoiseGit and your language is missing, download and install a matching language pack now and hit "Refresh" before continuing.
Language:
English
Refresh
Download language packs:
https://tortoisegit.org/download/
< 上一步(B)
下一步(N) >
取消
帮助


First Start Wizard - TortoiseGit
Welcome to TortoiseGit!
TortoiseGit is designed as a Shell Extension. Therefore the main interaction with TortoiseGit will be using the context menu of the Windows shell such as the Windows Explorer.
This is the First Start Wizard which will help you configure the basic settings. TortoiseGit is highly configurable, so it is advisable that you have a look into its settings dialog. The settings dialog is reachable using the start menu or the context menu of the Windows Explorer: TortoiseGit -> Settings.
Please see the manual for general hints. E.g. in most dialogs there are some shared shortcuts such as F5 for refresh or CTRL+F for search and oftentimes there are powerful context menus.
We hope you enjoy using TortoiseGit! In case of questions or problems, please consult our manual or go to https://tortoisegit.org/support/
< 上一步(B)
下一步(N) >
取消
帮助



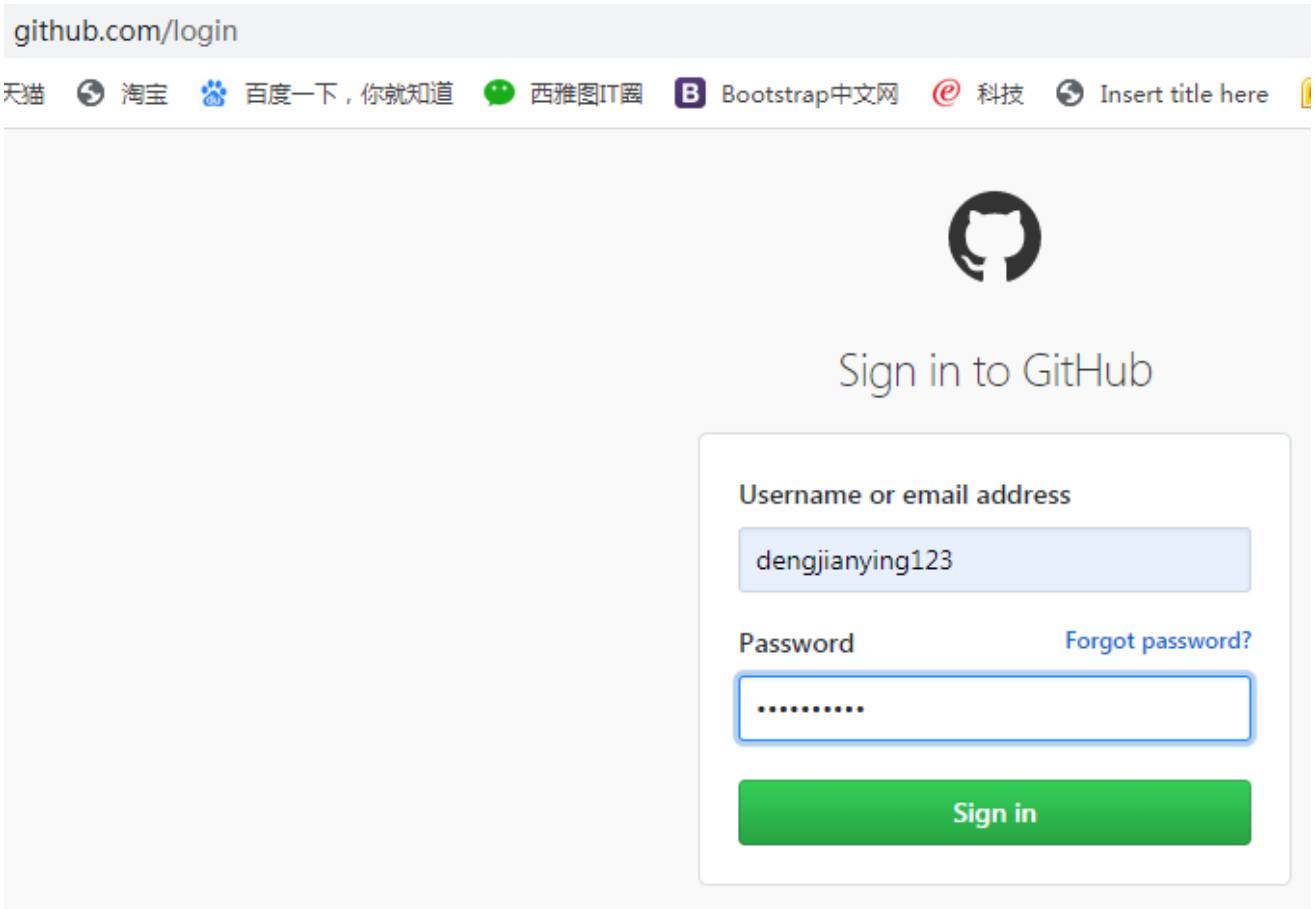
下图输入前面注册过的github的账号与邮箱地址：



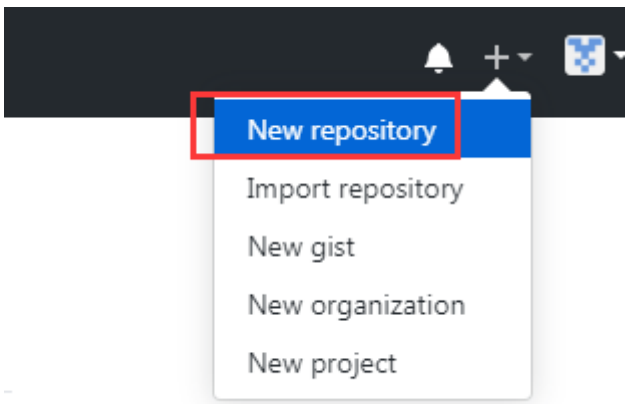


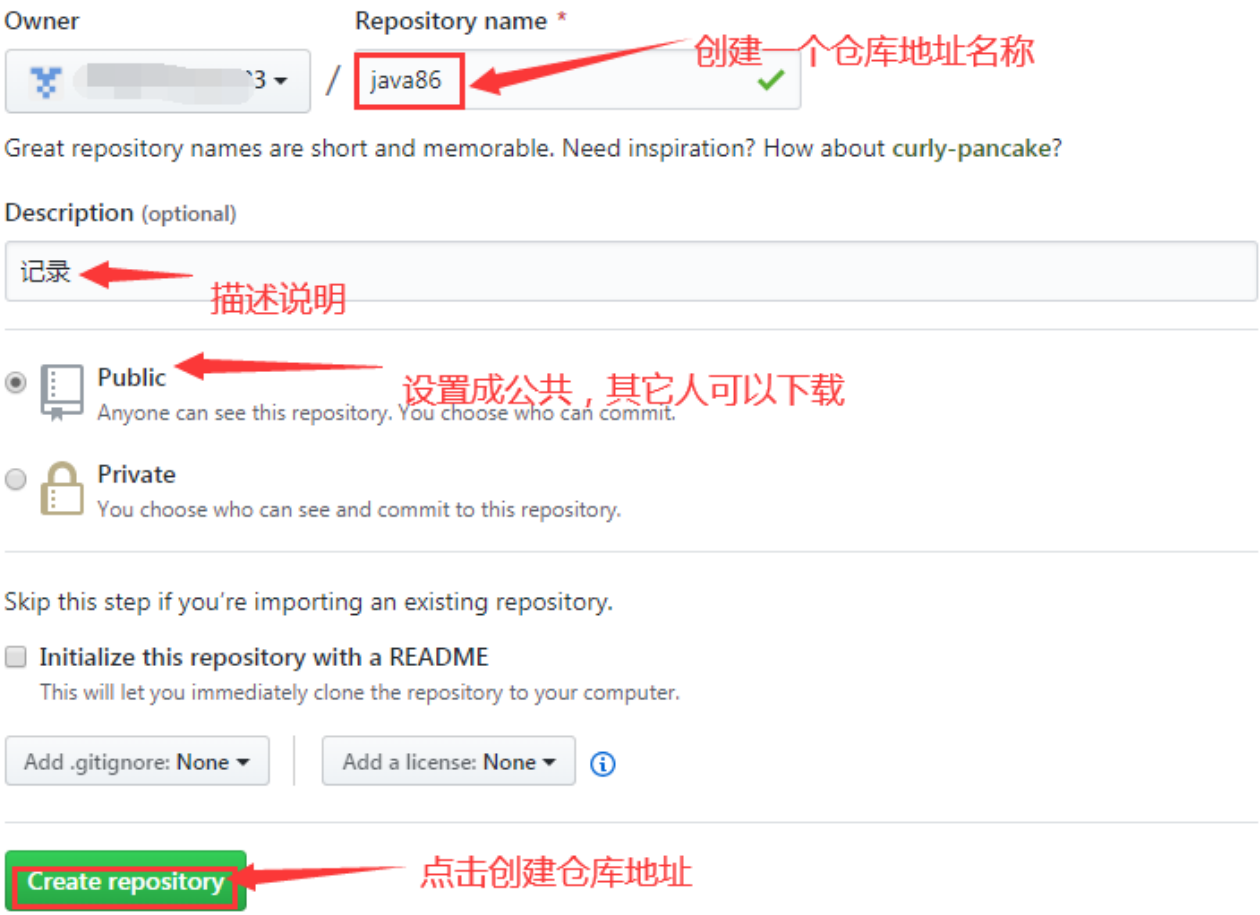
上传至开源平台github

浏览器打开github平源开台https://github.com/login使用前面注册过的账号进行登录：

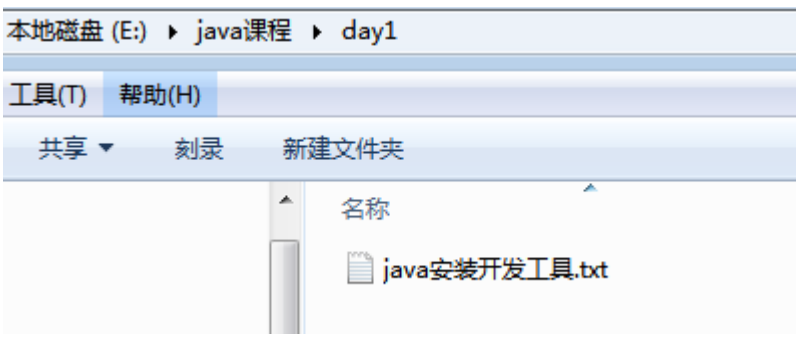


进入github,创建新仓库地址：

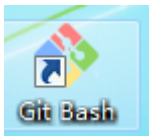




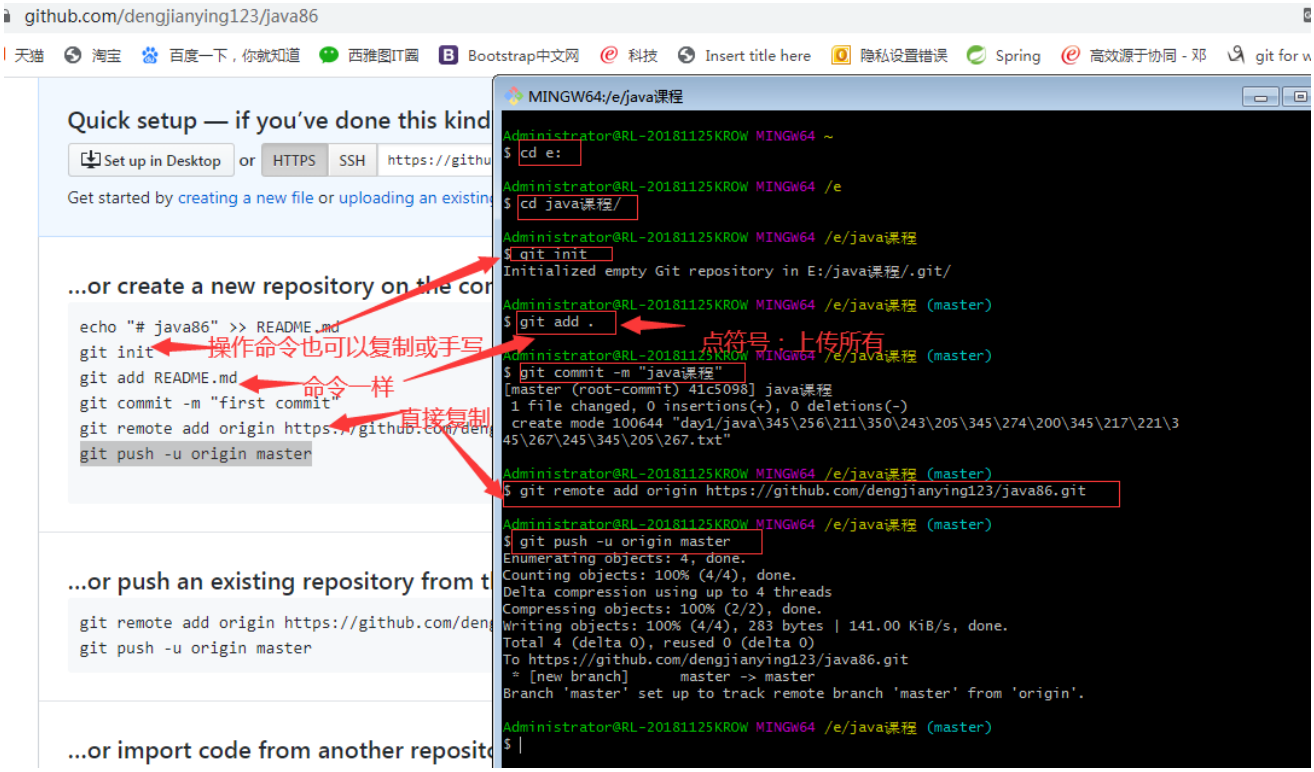
假如笔记本电脑上有一个文件夹如：java课程，里面有day1文件夹与文件等需要上传到刚才在github创建的java86这个下面：



打开Git Bash,如图:



打开服务端Git Bash命令操作上传：



git上传操作命令：

```
Administrator@RL-20181125KROW MINGW64 ~
$ cd e:

Administrator@RL-20181125KROW MINGW64 /e
$ cd java课程/

Administrator@RL-20181125KROW MINGW64 /e/java课程
$ git init
Initialized empty Git repository in E:/java课程/.git/

Administrator@RL-20181125KROW MINGW64 /e/java课程 (master)
$ git add .

Administrator@RL-20181125KROW MINGW64 /e/java课程 (master)
$ git commit -m "java课程"
[master (root-commit) 41c5098] java课程
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644
"day1/java\345\256\211\350\243\205\345\274\200\345\217\221\345\267\245\345\205\267.txt"
```

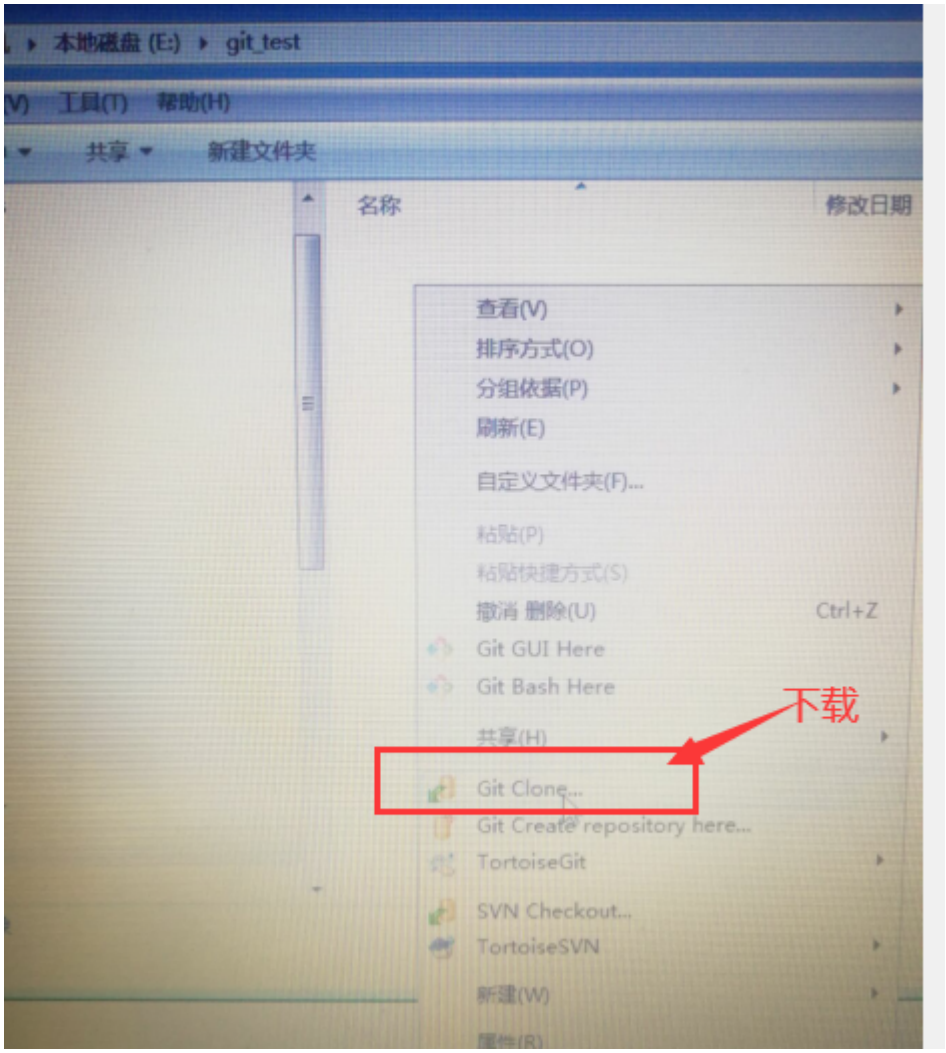
```
Administrator@RL-20181125KROW MINGW64 /e/java课程 (master)
$ git remote add origin https://github.com/dengjianying123/java86.git

Administrator@RL-20181125KROW MINGW64 /e/java课程 (master)
$ git push -u origin master
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 283 bytes | 141.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0)
To https://github.com/dengjianying123/java86.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.

Administrator@RL-20181125KROW MINGW64 /e/java课程 (master)
$
```

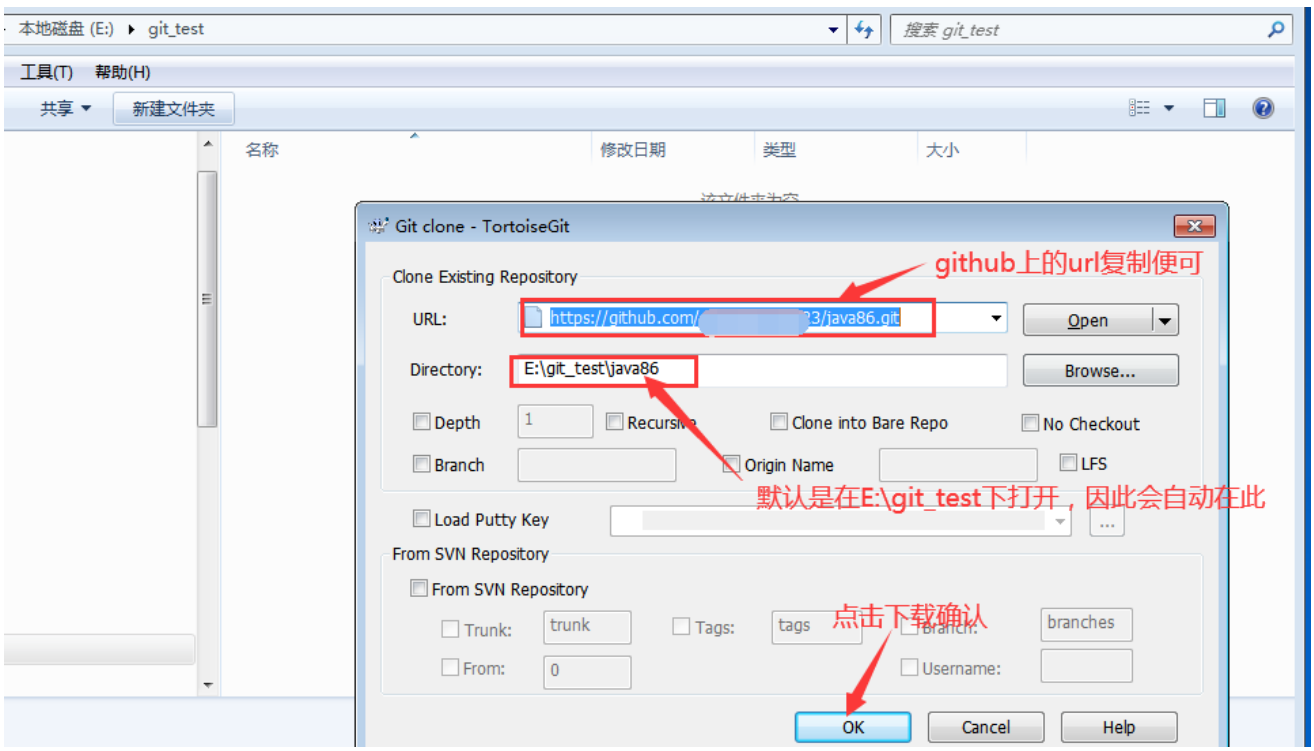
上次如果要上传文件，只需要在客户端：git add ,然后点击commit and push便可。

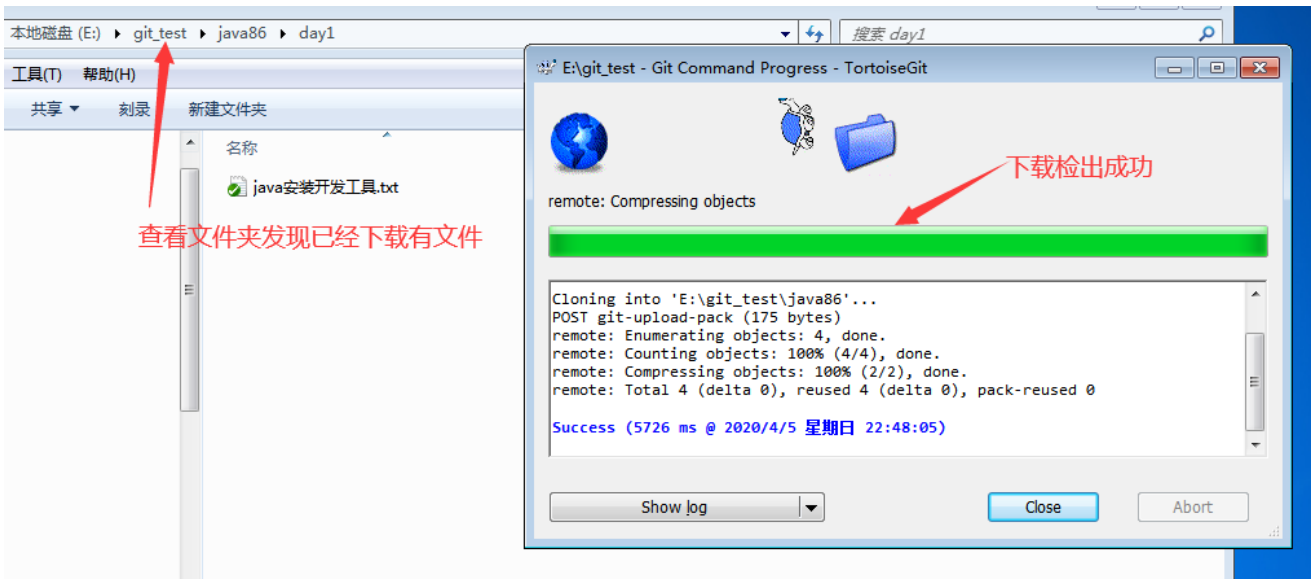
查看下载Git Clone文件

笔记本电脑任意盘符创建一个准备要下载的文件夹，如在E盘创建git_test：

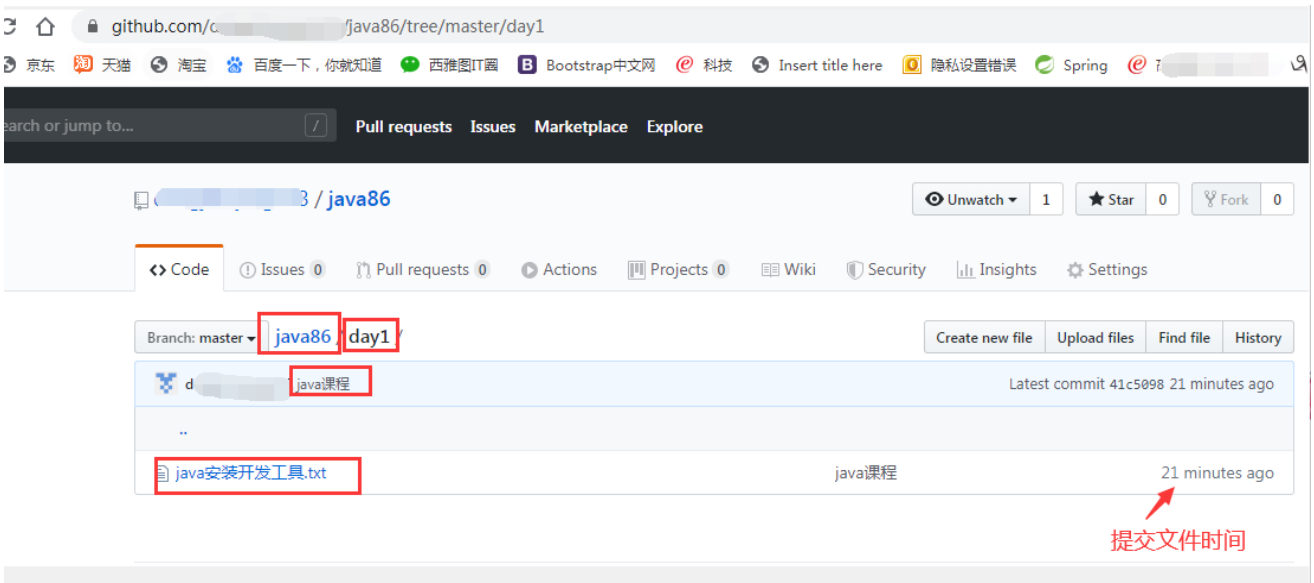
打开准备要下载文件的git_test文件夹，右击，选择**Git Clone**:


本地磁盘 (E:) ▸ git_test
工具(T) 帮助(H)
共享 新建文件夹
名称
修改日期
查看(V)
排序方式(O)
分组依据(P)
刷新(E)
自定义文件夹(F)...
粘贴(P)
粘贴快捷方式(S)
撤消 删除(U) Ctrl+Z
Git GUI Here
Git Bash Here
共享(H)
Git Clone...
Git Create repository here...
TortoiseGit
SVN Checkout...
TortoiseSVN
新建(W)
下载


本地磁盘 (E:) ▸ git_test
搜索 git_test
工具(T) 帮助(H)
共享 新建文件夹
名称
修改日期
类型
大小
Git clone - TortoiseGit
Clone Existing Repository
URL:
https://github.com/ 3/java86.git
Open
Directory:
E:\git_test\java86
Browse...
Depth
1
Recursive
Clone into Bare Repo
No Checkout
Branch
Origin Name
LFS
Load Putty Key
From SVN Repository
From SVN Repository
Trunk:
trunk
Tags:
tags
branches
From:
0
Username:
OK
Cancel
Help
github上的url复制便可
默认是在E:\git_test下打开，因此会自动在此
点击下载确认

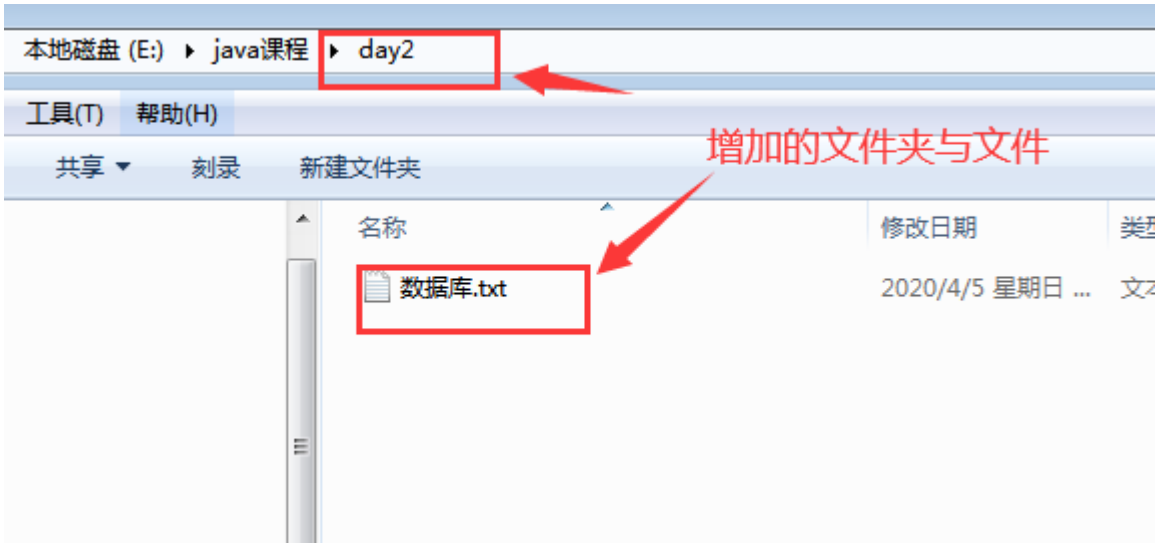


查看github平台：

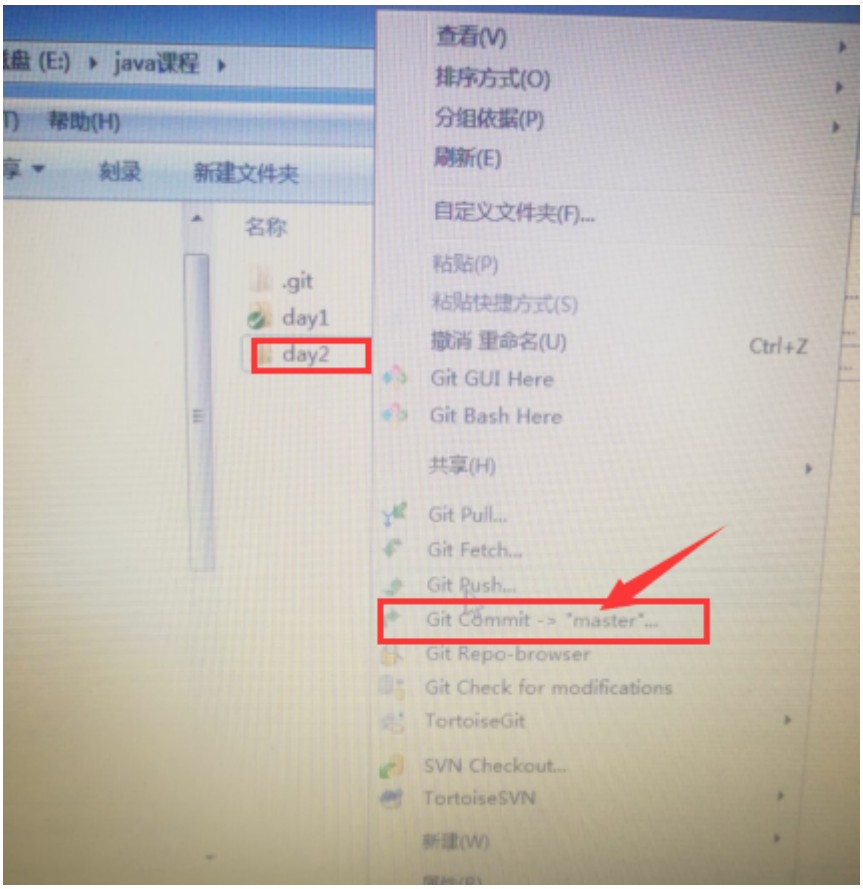


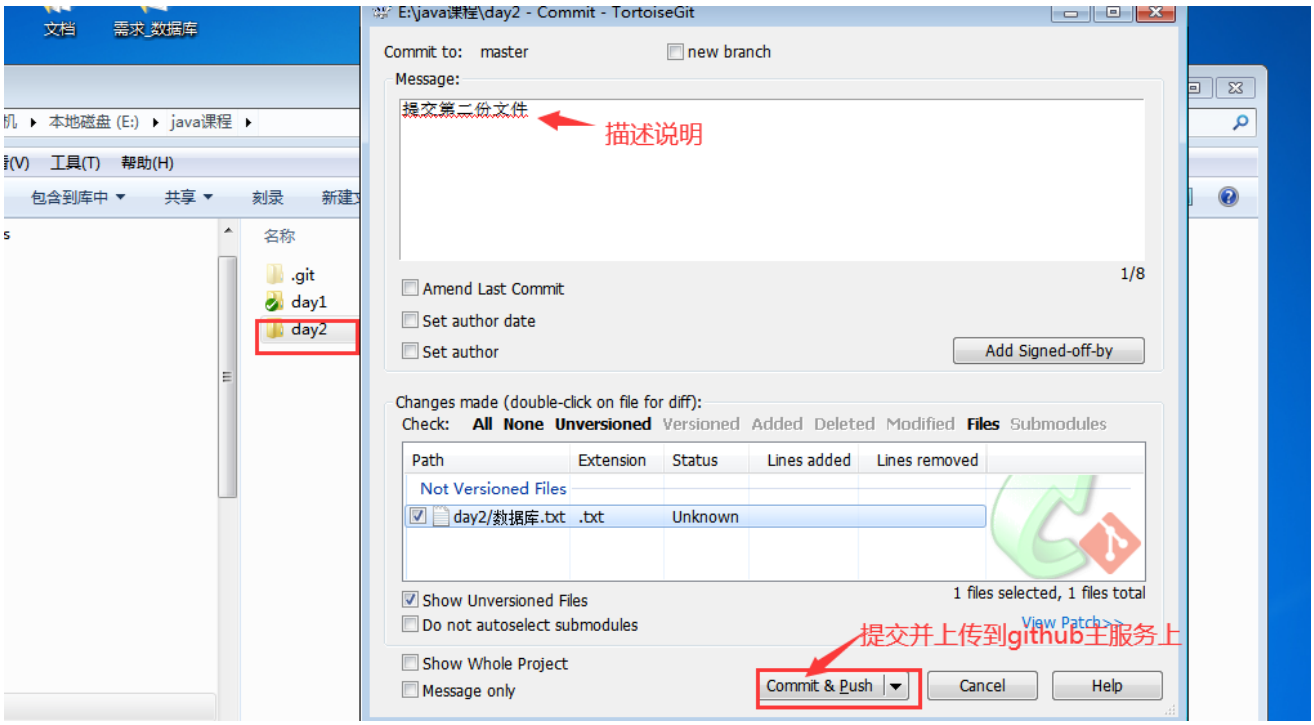
补充文件git commit上传

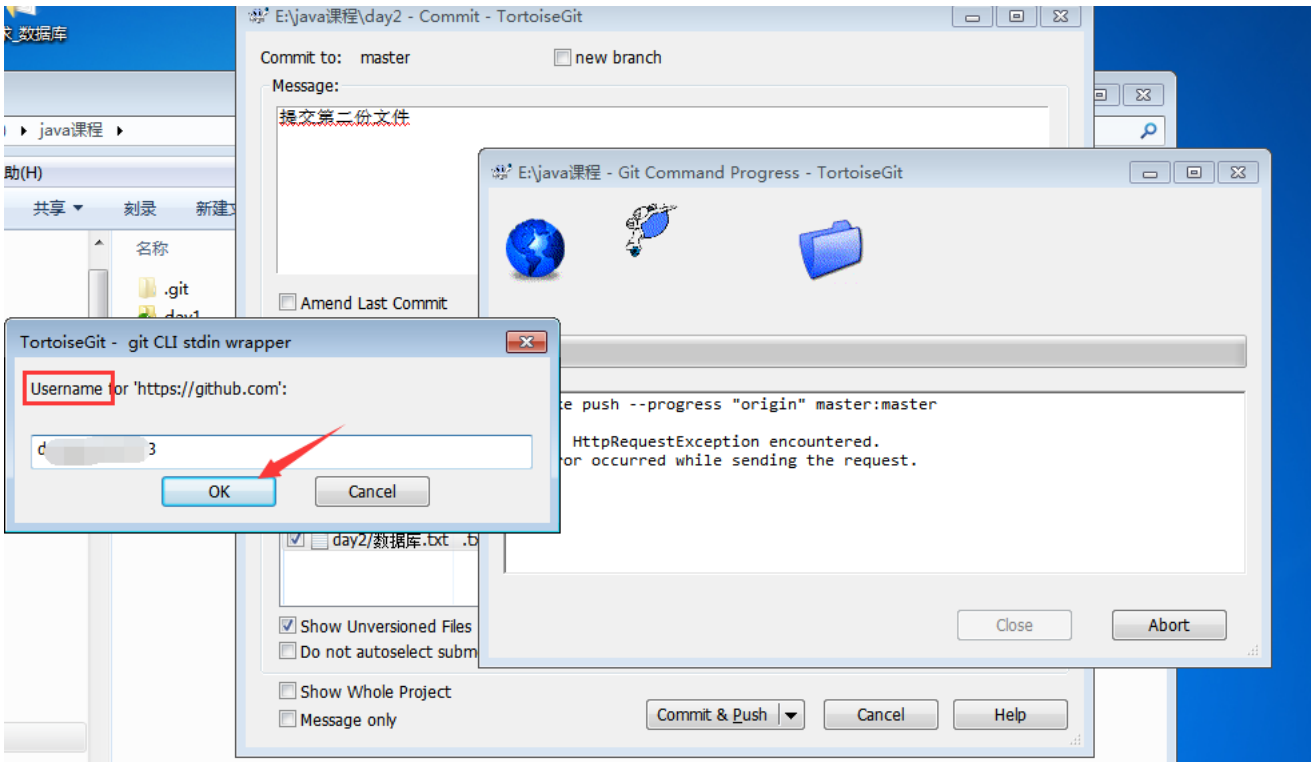
如果再需要上传文件或是文件夹

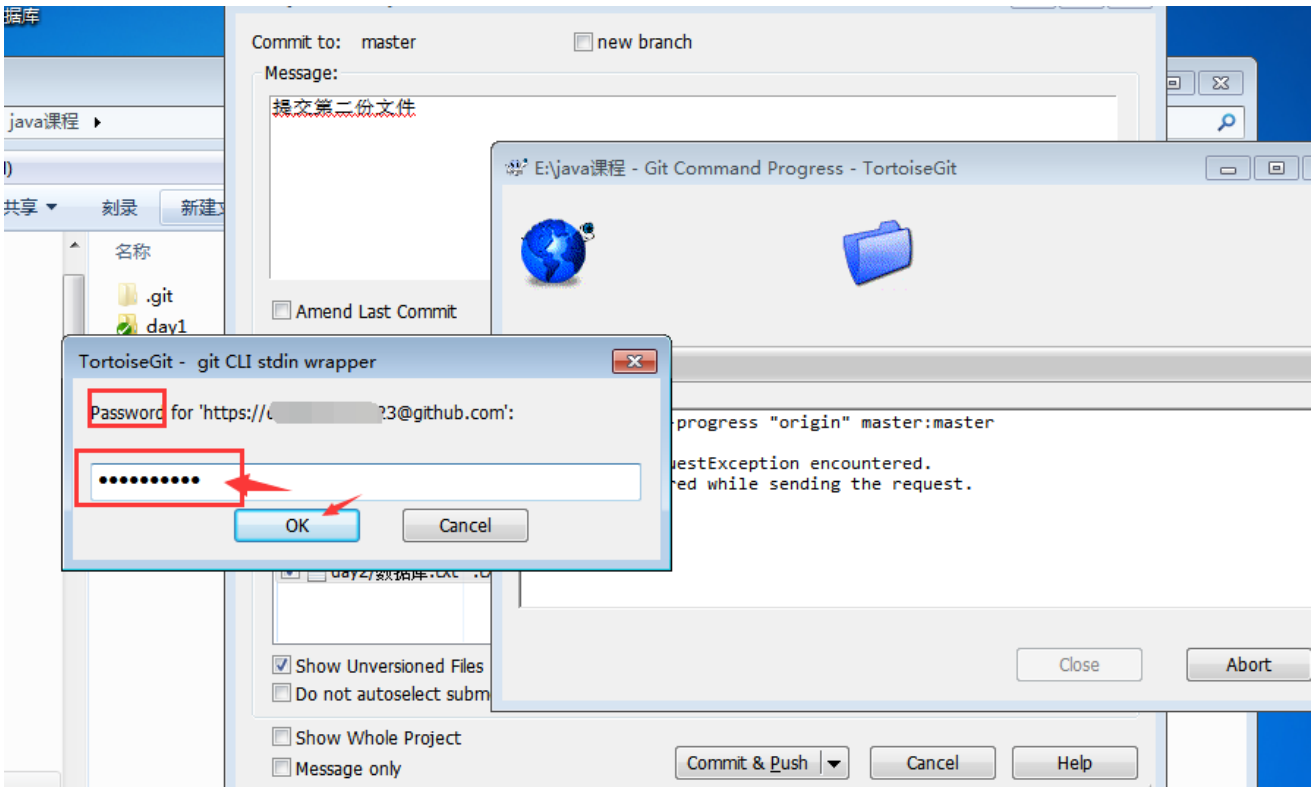


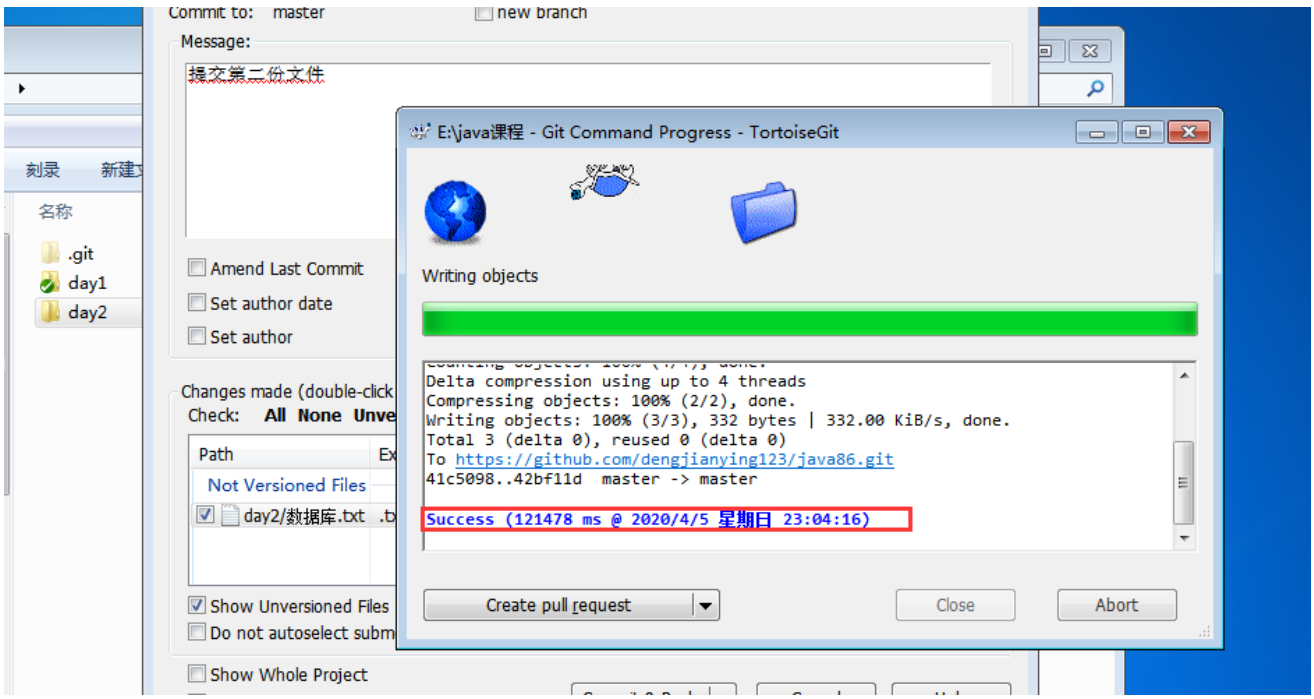
在java课程选中day2这个文件夹，右击，选择git commit-master便可：

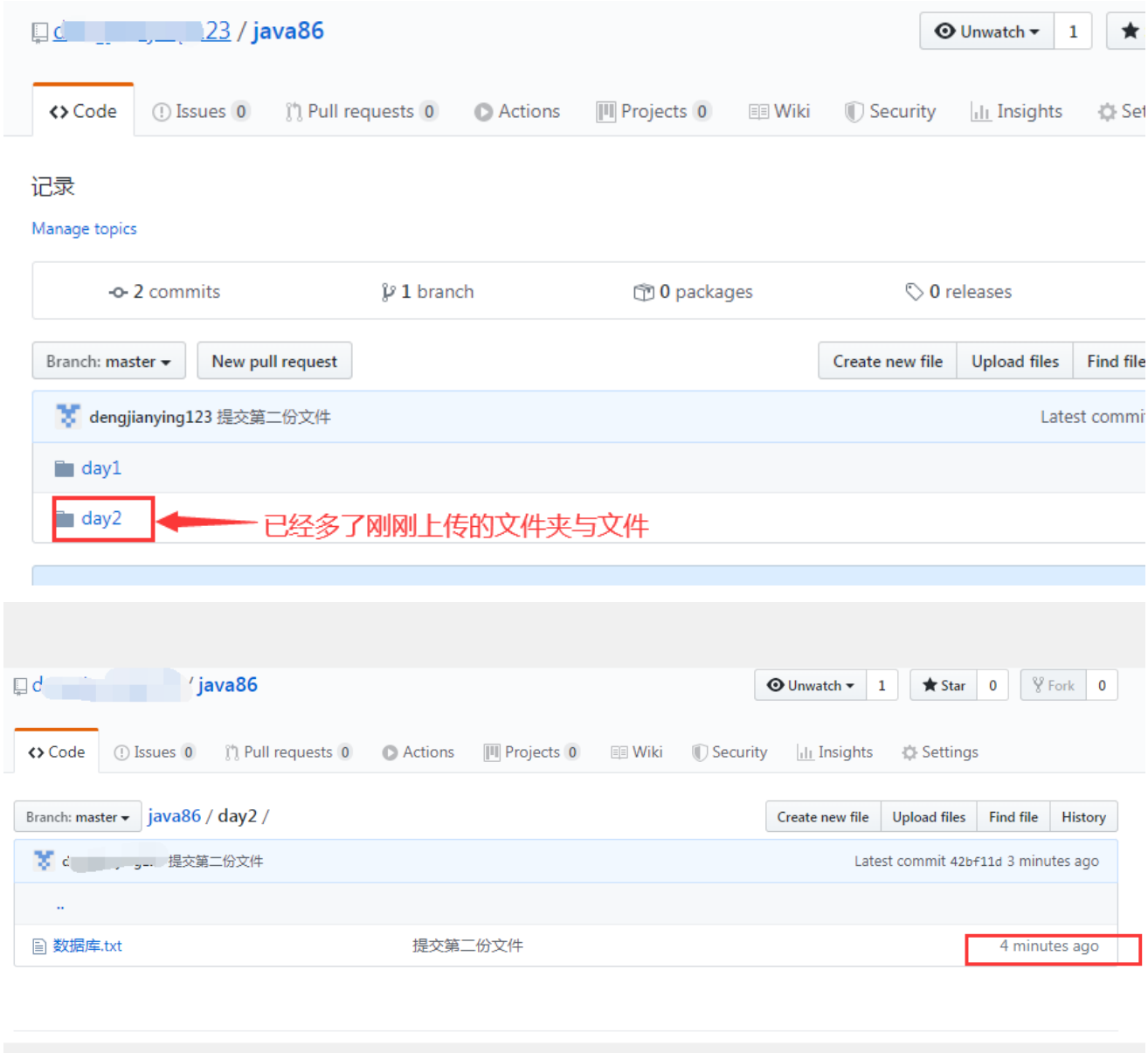



E:\java课程\day2 - Commit - TortoiseGit
Commit to: master
new branch
Message:
提交第二份文件
描述说明
1/8
Amend Last Commit
Set author date
Set author
Add Signed-off-by
Changes made (double-click on file for diff):
Check: All None Unversioned Versioned Added Deleted Modified Files Submodules
Path
Extension
Status
Lines added
Lines removed
Not Versioned Files
day2/数据库.txt .txt Unknown
1 files selected, 1 files total
Show Unversioned Files
Do not autoselect submodules
View Patch>>
提交并上传到github主服务上
Show Whole Project
Message only
Commit & Push
Cancel
Help
本地磁盘 (E:) ▸ java课程 ▸
工具(T) 帮助(H)
包含到库中
共享
刻录
名称
.git
day1
day2


E:\java课程\day2 - Commit - TortoiseGit
Commit to: master
new branch
Message:
提交第二份文件
E:\java课程 - Git Command Progress - TortoiseGit
TortoiseGit - git CLI stdin wrapper
Username for 'https://github.com':
OK
Cancel
e push --progress "origin" master:master
HttpRequestException encountered.
or occurred while sending the request.
Close
Abort
Amend Last Commit
day2/数据库.txt
Show Unversioned Files
Show Whole Project
Message only
Commit & Push
Cancel
Help

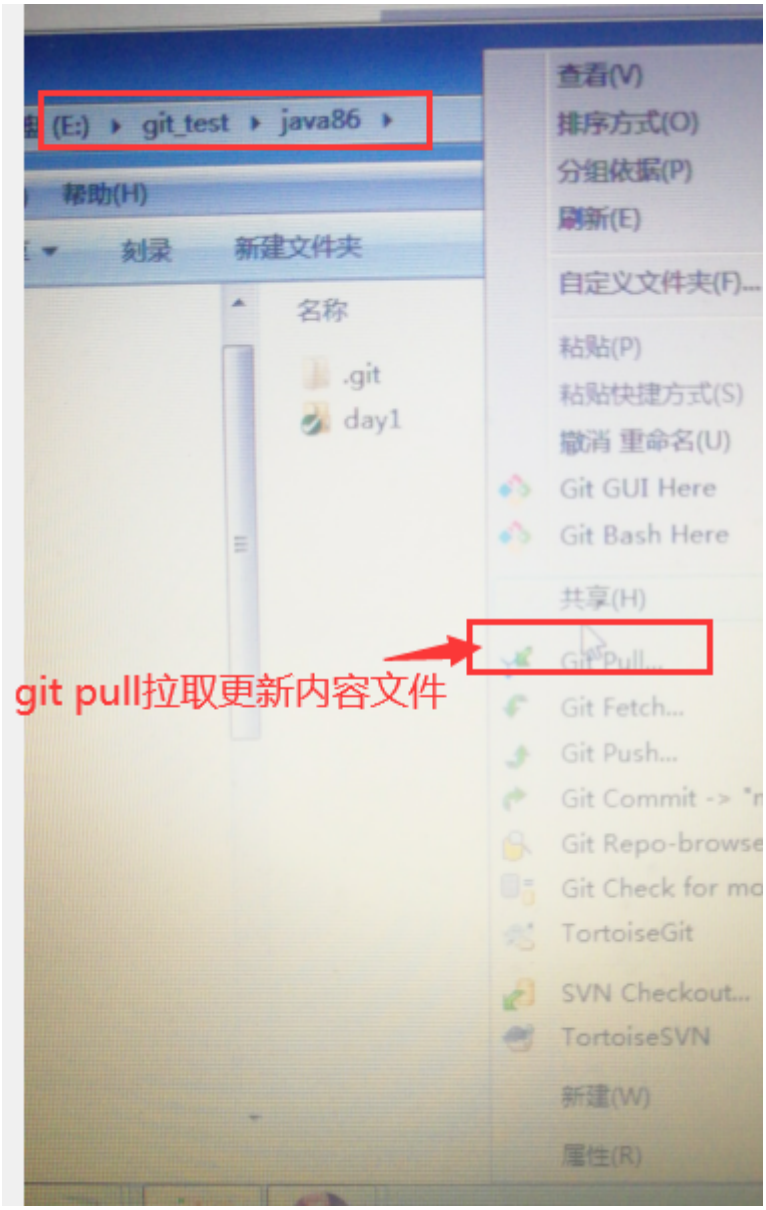

Commit to: master
new branch
Message:
提交第二份文件
E:\java课程 - Git Command Progress - TortoiseGit
Amend Last Commit
TortoiseGit - git CLI stdin wrapper
Password for 'https://
?3@github.com':
OK
Cancel
progress "origin" master:master
estException encountered.
ed while sending the request.
Close
Abort
Show Unversioned Files
Do not autoselect subm
Show Whole Project
Message only
Commit & Push
Cancel
Help

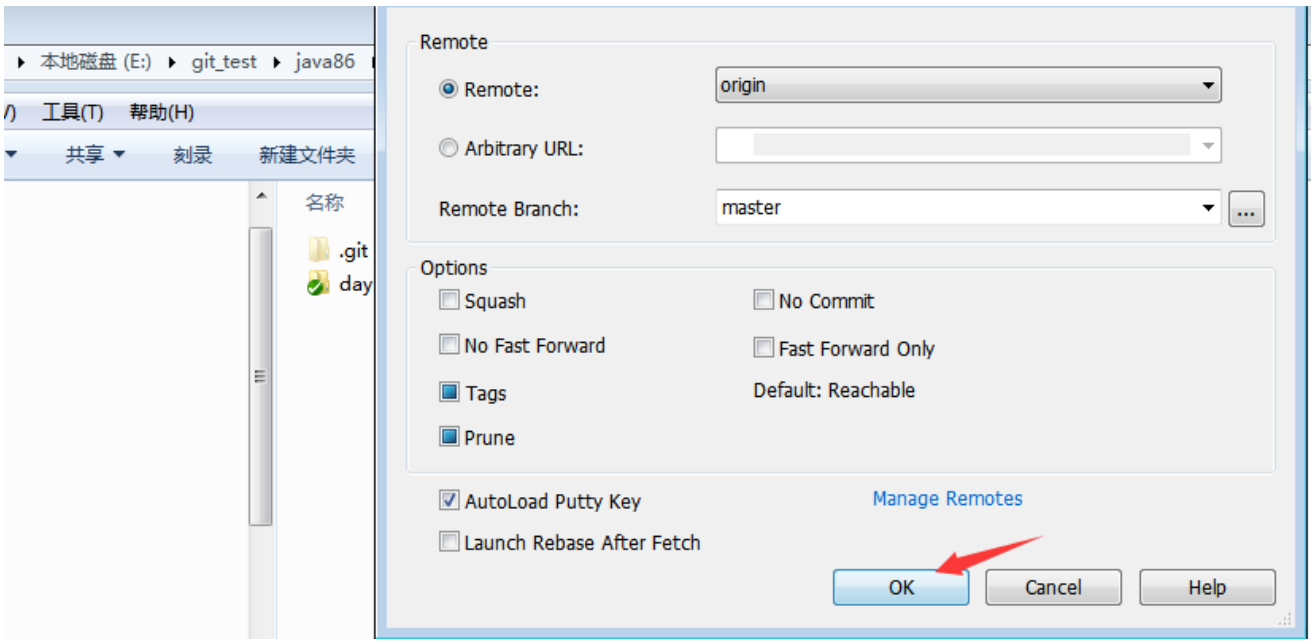

Commit to: master
new branch
Message:
提交第二份文件
E:\java课程 - Git Command Progress - TortoiseGit
Writing objects
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 332 bytes | 332.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/dengjianying123/java86.git
41c5098..42bf11d master -> master
Success (121478 ms @ 2020/4/5 星期日 23:04:16)
Create pull request
Close
Abort
.git
day1
day2
Amend Last Commit
Set author date
Set author
Changes made (double-click
Check: All None Unve
Path
Not Versioned Files
day2/数据库.txt
Show Unversioned Files
Do not autoselect subm
Show Whole Project

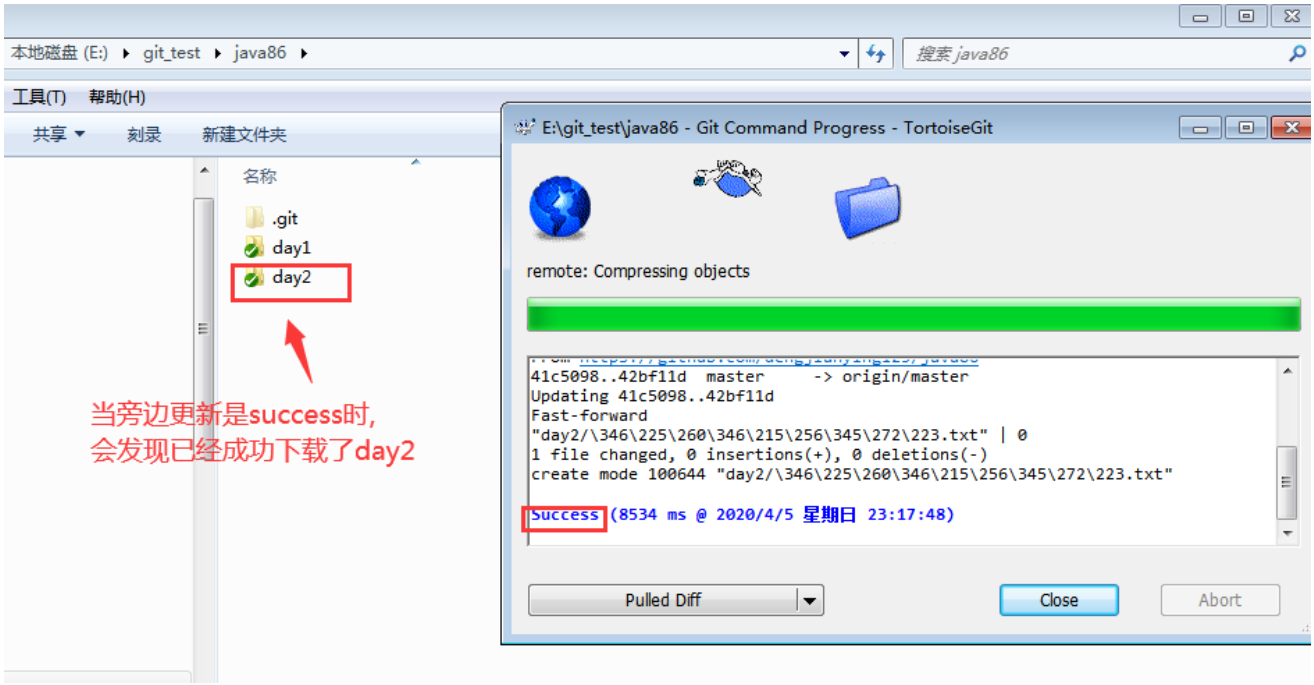


本地Git Pull下载

如要更新最新文件，打开git_test这个文件夹：任意空白右击选择Git Pull,会看到一个拉下来的绿色图标标示


(E:) ▸ git_test ▸ java86 ▸
帮助(H)
刻录
新建文件夹
名称
.git
day1
查看(V)
排序方式(O)
分组依据(P)
刷新(E)
自定义文件夹(F)...
粘贴(P)
粘贴快捷方式(S)
撤消 重命名(U)
Git GUI Here
Git Bash Here
共享(H)
Git Pull...
Git Fetch...
Git Push...
Git Commit -> "r
Git Repo-browse
Git Check for mo
TortoiseGit
SVN Checkout...
TortoiseSVN
新建(W)
属性(R)
git pull拉取更新内容文件


本地磁盘 (E:) ▸ git_test ▸ java86
工具(T) 帮助(H)
共享
刻录
新建文件夹
名称
.git
day
Remote
Remote:
origin
Arbitrary URL:
Remote Branch:
master
Options
Squash
No Commit
No Fast Forward
Fast Forward Only
Tags
Default: Reachable
Prune
AutoLoad Putty Key
Manage Remotes
Launch Rebase After Fetch
OK
Cancel
Help


本地磁盘 (E:) ▸ git_test ▸ java86 ▸
搜索 java86
工具(T) 帮助(H)
共享
刻录
新建文件夹
名称
.git
day1
day2
当旁边更新是success时,
会发现已经成功下载了day2
E:\git_test\java86 - Git Command Progress - TortoiseGit
remote: Compressing objects
41c5098..42bf11d master -> origin/master
Updating 41c5098..42bf11d
Fast-forward
"day2/\346\225\260\346\215\256\345\272\223.txt" | 0
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 "day2/\346\225\260\346\215\256\345\272\223.txt"
Success (8534 ms @ 2020/4/5 星期日 23:17:48)
Pulled Diff
Close
Abort