

1 Gitlab 账号注册/登录流程

1.1 访问地址

URL: <https://code.geosts.org>

1.2 注册

使用邮箱进行注册，设置用户名和密码



GitLab Community Edition

Open source software to collaborate on code

Manage Git repositories with fine-grained access controls that keep your code secure. Perform code reviews and enhance collaboration with merge requests. Each project can also have an issue tracker and a wiki.

Sign in	Register
Full name	
<input type="text"/>	
Username	
<input type="text"/>	
Email	
<input type="text"/>	
Email confirmation	
<input type="text"/>	
Password	
<input type="password"/>	
Minimum length is 8 characters	
<input type="button" value="Register"/>	

1.3 登录

使用用户名或邮箱进行登录



GitLab Community Edition

Open source software to collaborate on code

Manage Git repositories with fine-grained access controls that keep your code secure. Perform code reviews and enhance collaboration with merge requests. Each project can also have an issue tracker and a wiki.

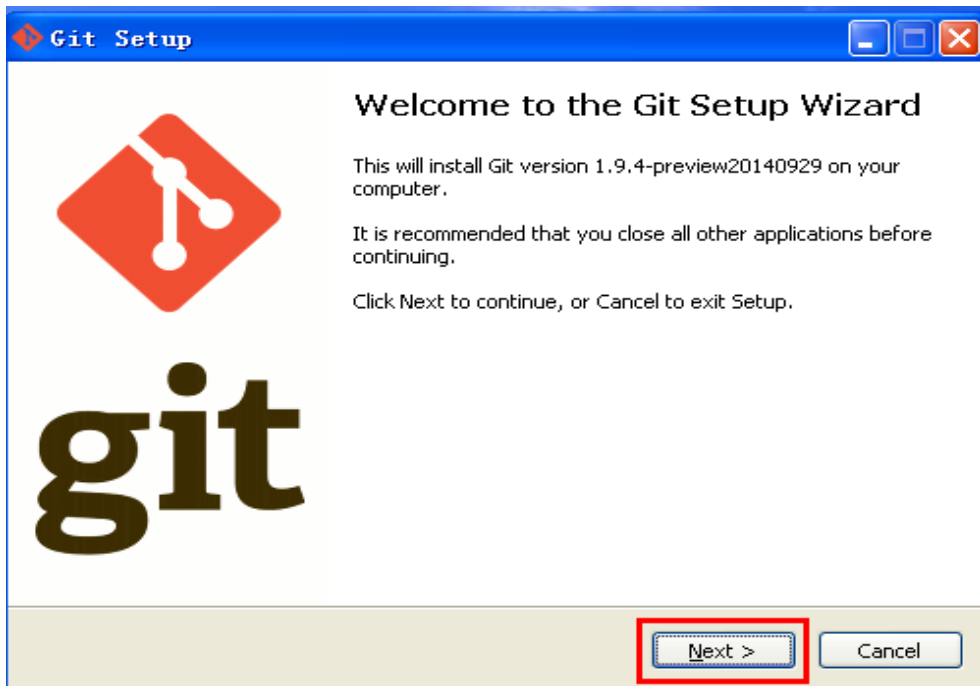
Sign in	Register
Username or email	
<input type="text"/>	
Password	
<input type="password"/>	
<input type="checkbox"/> Remember me	
Forgot your password?	
<input type="button" value="Sign in"/>	

2 Git 环境搭建（Windows）

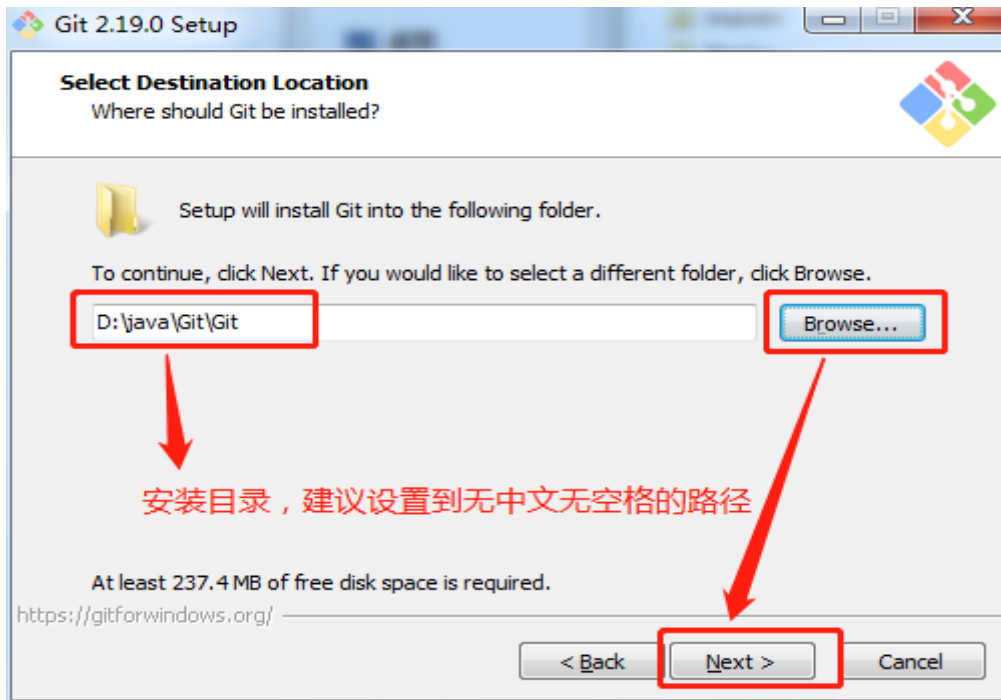
2.1 安装 GIT

1) 下载 Git 安装包: <https://git-scm.com/downloads>

2) 运行安装文件, 点击 Next 进行安装:



3) 选择安装路径

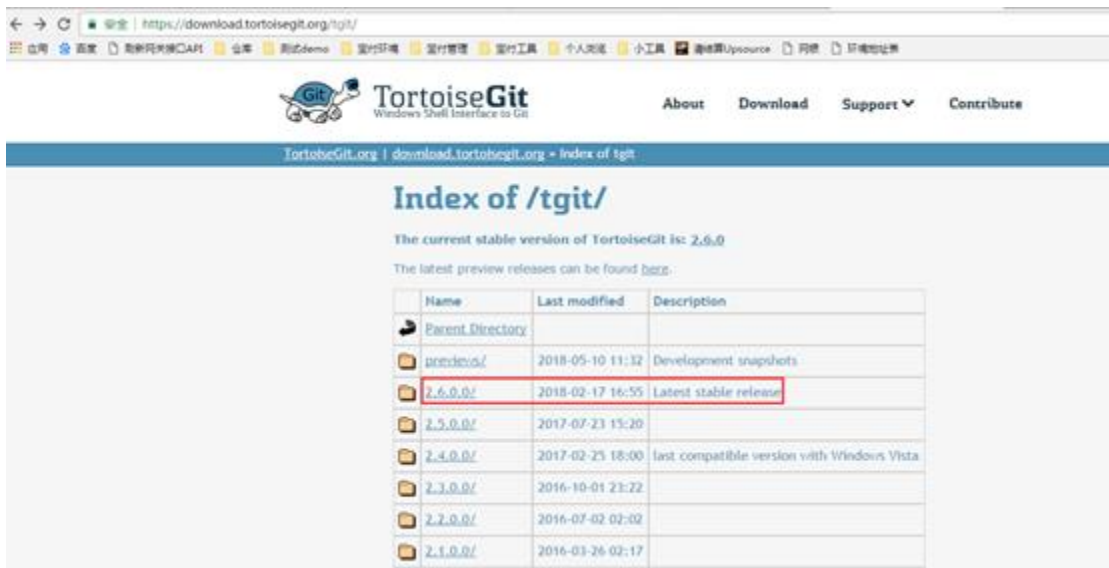


4) 一直 Next, 并完成安装。

2.2 安装 TortoiseGit

tortoiseGit 是一个开放的 git 版本控制系统的源客户端，支持 Winxp/vista/win7，该软件功能和 git 一样。不同的是：git 是命令行操作模式，tortoiseGit 界面化操作模式，不用记 git 相关命令就可以直接操作。

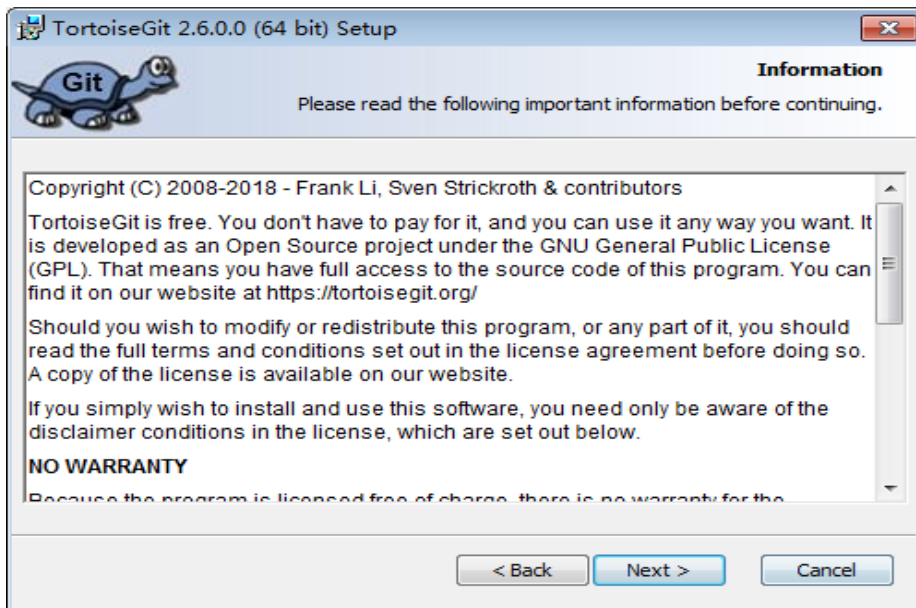
1) tortoiseGit 下载地址: <https://download.tortoisegit.org/tgit/>



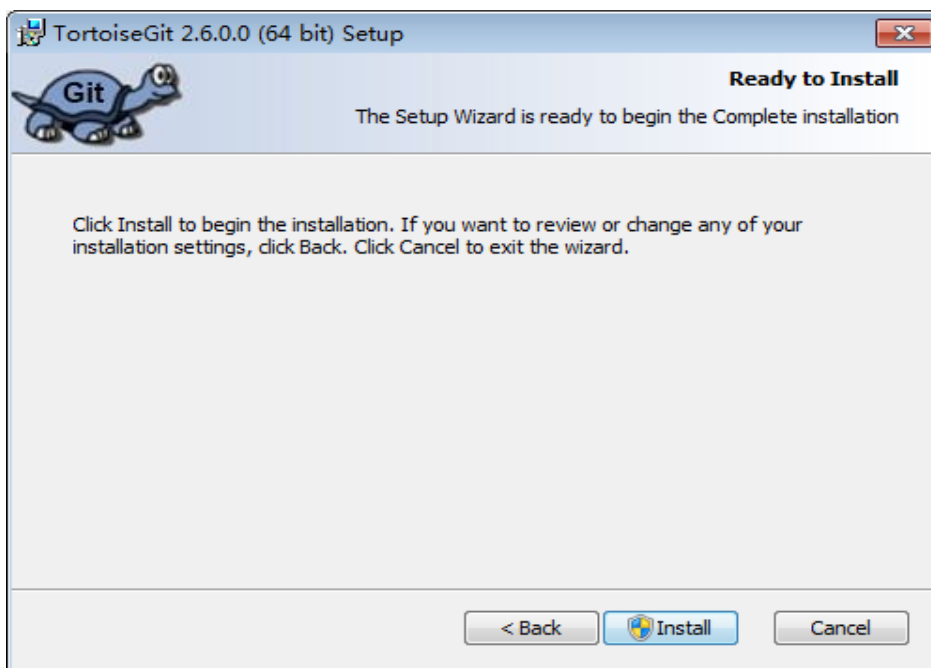
2) 运行 TortoiseGit-2.6.0.0-64bit.msi，弹出安装导向页面



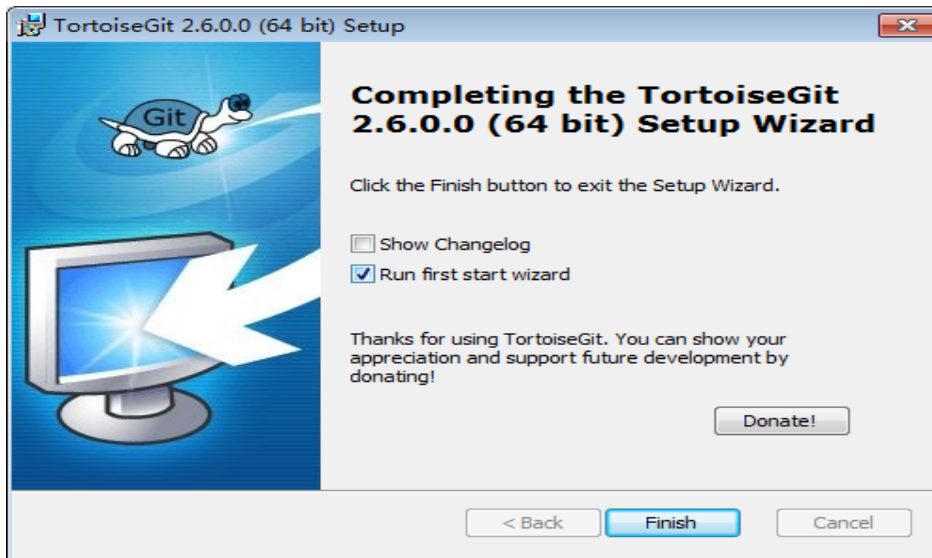
3) 一路 Next>，配置均选择默认配置



4) 点击 Install

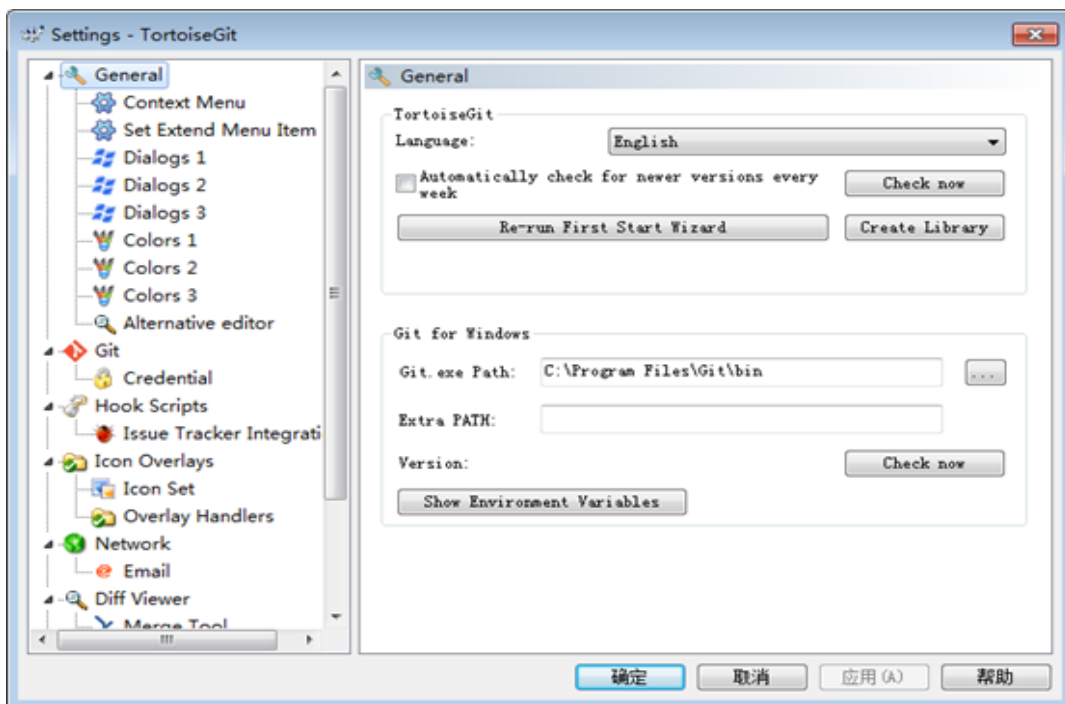


5) 点击 Finish, 如果以前有来版本, 则选择覆盖, 关闭就程序并尝试重启即可



2.3 TortoiseGit 配置

- 1) 在空白处点击鼠标右键，选择 --> TortoiseGit --> Settings，弹出配置界面(当 TortoiseGit 安装完成后，鼠标右键点击后，默认出现 TortoiseGit 相关选项)



- 2) 点击左边 Git, 配置相关信息，配置好后保存退出

因为当前还没有本地项目,所以 “编辑本地 .git/config(L)” 按钮处于灰色不可用状态,如果在某个本地 Git 项目下打开配置对话框,那么这个按钮就可用,然后就可以编辑此项目的一些属性。

点击 “编辑全局 .git/config(0)” 按钮,会使用记事本打开全局配置文件,在全局配置文件中,在后面加上下面的内容:

```
[user]
  name = bluethink
  email = bt@qq.com
  signingkey = 123456
[credential]
  helper = store
```

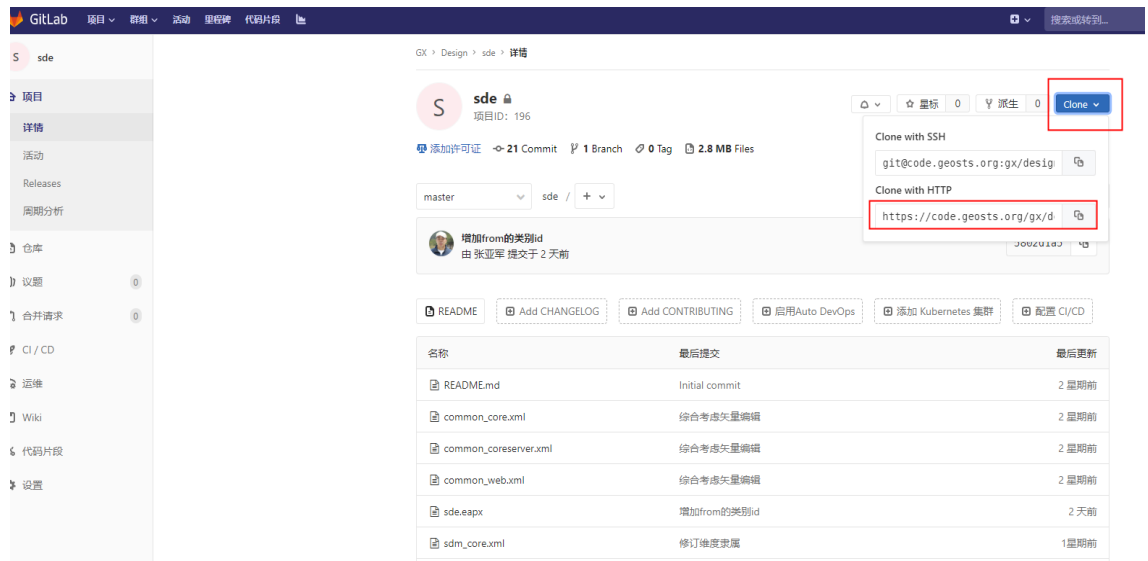
则当你推送项目到 Gitlab 等在线仓库时,会记住你输入的用户名和密码(这里不是用户的姓名和 Email),这里用户名和密码就是注册 gitlab 的用户名和密码。

3 TortoiseGit 用法

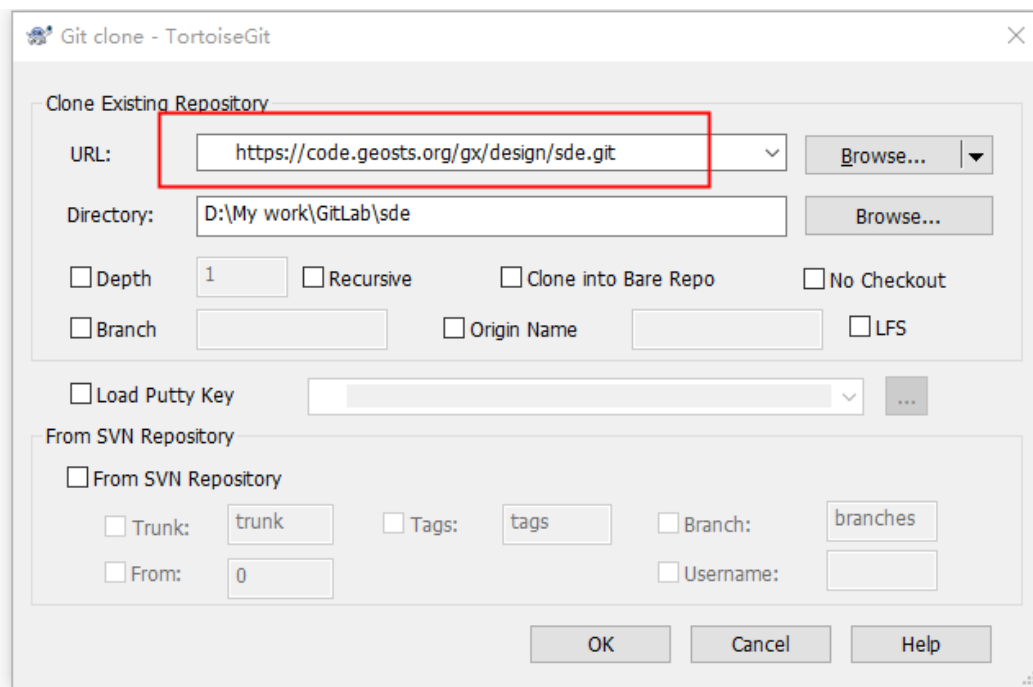
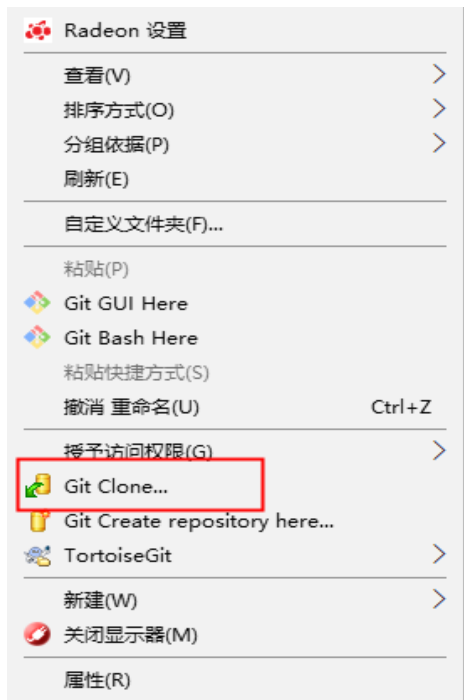
3.1 项目 Clone

将线上的项目克隆到本地。

1) 找到需要克隆的项目，点击克隆按钮，复制 HTTP 地址

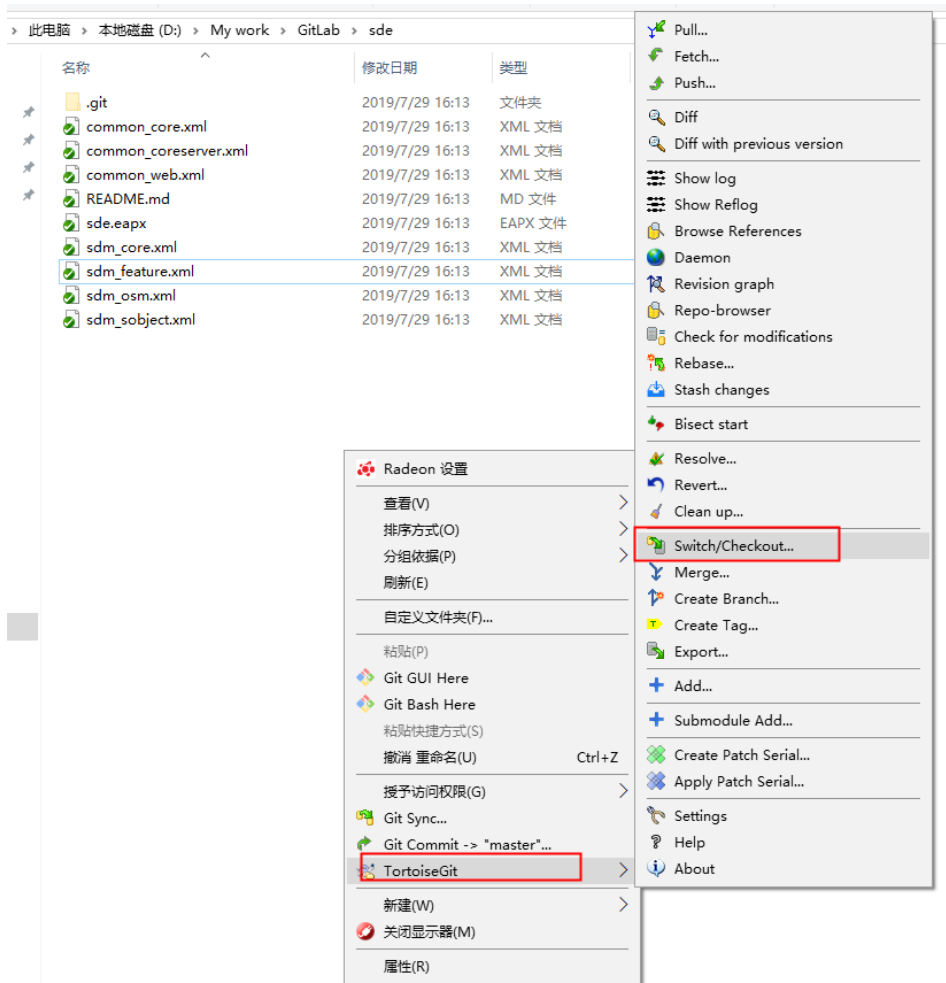


2) 在本地选择项目存放地址，空白处右键点击，选择 Git Clone...，将复制的 HTTP 地址粘贴在 URL 中，点击 OK

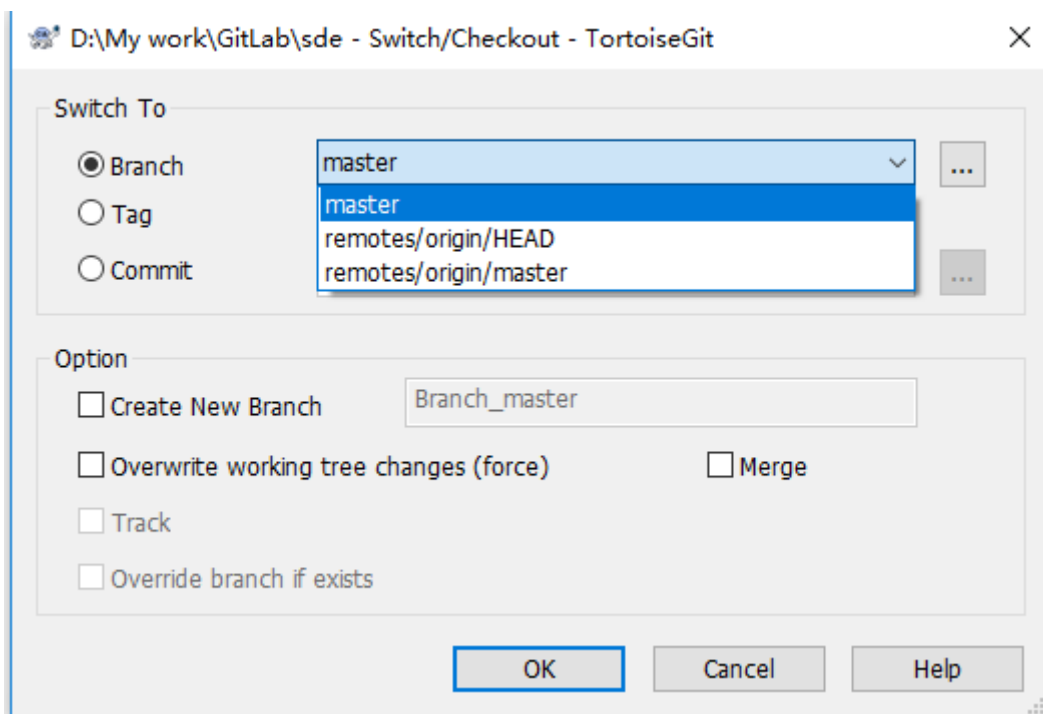


3.2 分支切换

打开克隆下来的项目，右键点击空白处，选择 TortoiseGit->Switch/Checkout...

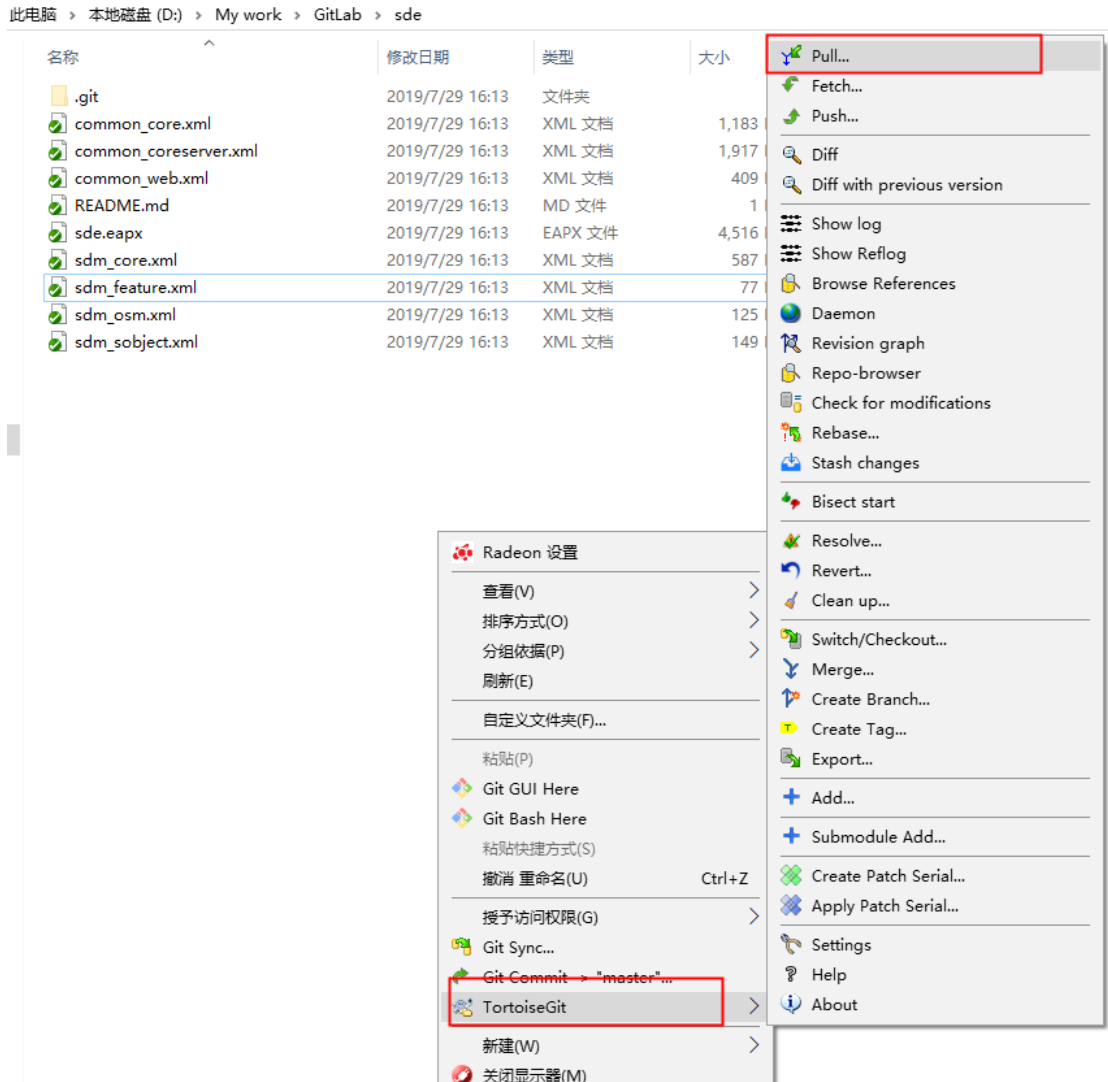


选择所需要操作的分支，点击 OK



3.3 项目 Pull

打开克隆下来的项目，右键点击空白处，选择 Tortoise Git—>Pull，拉取代码并合并到本地仓库当前分支。



如果出现冲突，则先解决冲突，然后点击 Resolve 进行解决。

功能介绍：

Pull，拉取线上代码并进行合并

Fetch，拉取线上代码不进行合并

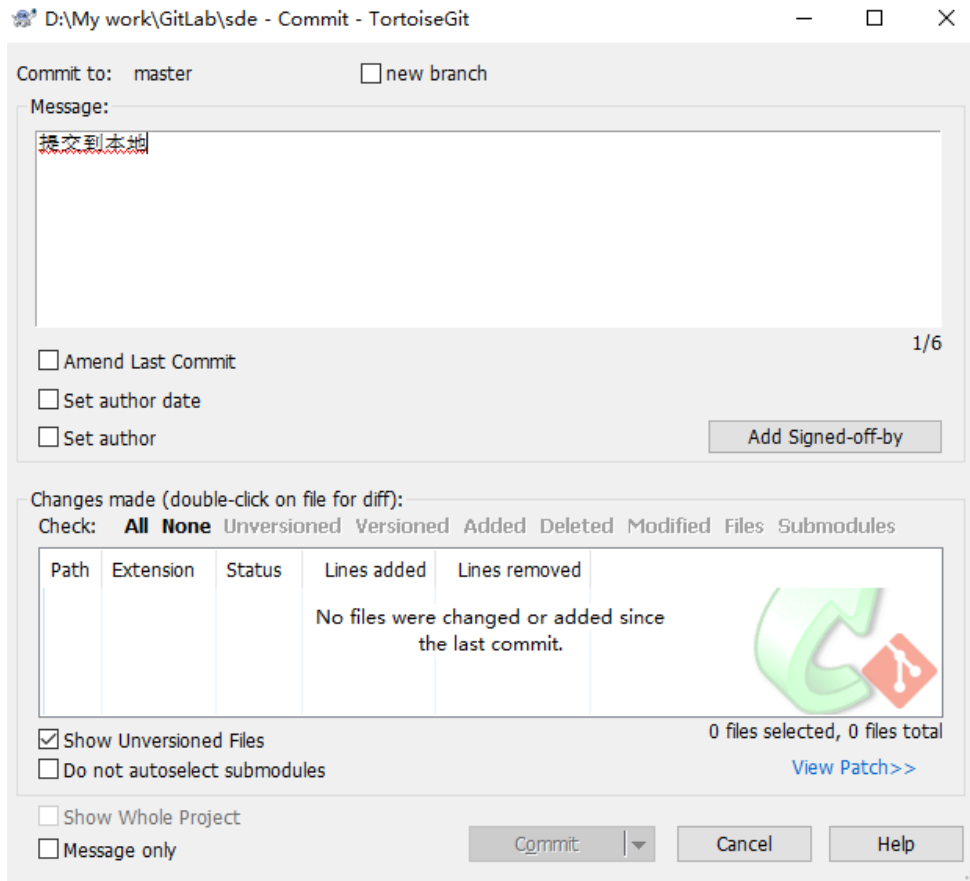
3.4 项目 commit 至本地

当研发任务没有完成，需要后续继续开发时，可以将代码提交至本地仓库进行版本控制。

打开所在项目，右键点击空白处，选择 Git Commit->"分支名称"。



在提交页面填写提交信息，用来记录本次提交所修改的内容



3.5 项目 Push 至线上

当完成某项研发任务或是一个完成的功能模块时，则需要将研发成果推到线上，这样别人才可以在线上查看所写内容。

打开所在项目，右键点击空白处，Tortoise Git—> push

期	类型	大小
7/29 16:27	文件夹	
7/29 16:13	XML 文档	1,183 KB
7/29 16:13	XML 文档	1,917 KB
7/29 16:13	XML 文档	409 KB
7/29 16:13	MD 文件	1 KB
7/29 16:13	EAPX 文件	4,516 KB
7/29 16:13	XML 文档	587 KB
7/29 16:13	XML 文档	77 KB
7/29 16:13	XML 文档	125 KB
7/29 16:13	XML 文档	149 KB

- Pull...
- Fetch...
- Push...

- Diff
- Diff with previous version

- Show log
- Show Reflog
- Browse References
- Daemon
- Revision graph
- Repo-browser
- Check for modifications
- Rebase...
- Stash changes

- Bisect start

- Resolve...
- Revert...
- Clean up...

- Switch/Checkout...
- Merge...
- Create Branch...
- Create Tag...
- Export...

- Add...

- Submodule Add...

- Create Patch Serial...
- Apply Patch Serial...

- Settings
- Help
- About

- Radeon 设置

- 查看(V) >
- 排序方式(O) >
- 分组依据(P) >
- 刷新(E)

- 自定义文件夹(F)...

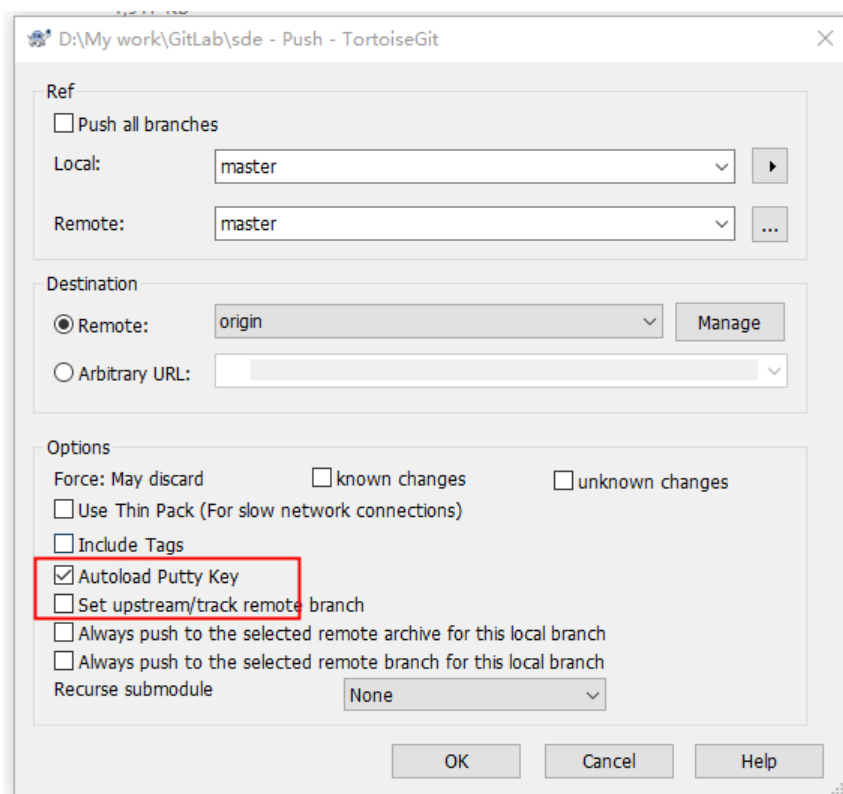
- 粘贴(P)
- Git GUI Here
- Git Bash Here
- 粘贴快捷方式(S)
- 撤销 重命名(U) Ctrl+Z

- 授予访问权限(G) >
- Git Sync...
- Git Commit -> "master"...
- TortoiseGit >

- 新建(W) >
- 关闭显示器(M)

- 属性(R)

在提交页面勾选 Autoload Putty Key, 即可将每次提交时记录的修改信息同时提交



至线上

5 项目管理

5.1 新建项目

新建项目时根据需求选择可见等级，当前群组中所有人具有新项目的权限

新建项目

项目可以用于存放文件(仓库)，安排计划(议题)，并发布文档(wiki)，及其他功能。

从模板或导入时为空白项目将启用所有功能，但可以在项目设置中将其禁用。

Information about additional Pages templates and how to install them can be found in our [Pages getting started guide](#).

提示：您也可以通过命令行来创建新项目。 [显示相关命令](#)

Blank project Create from template Import project

项目名称
My awesome project

项目 URL 项目标识串
https://code.geosts.org/ xhli my-awesome-project

Want to house several dependent projects under the same namespace? [Create a group](#).

Project description (optional)
Description format

Visibility Level [?](#)

- 私有
项目访问权限必须明确授权给每个用户。
- 内部
该项目允许已登录的用户访问。
- 公开
该项目允许任何人访问。

Initialize repository with a README
Allows you to immediately clone this project's repository. Skip this if you plan to push up an existing repository.

Create project Cancel

5.2 项目管理

5.2.1 分支

- 1) 项目主分支有且仅有一个，其他分支可以按照当前产品的版本号进行构建；
- 2) 研发人员需基于分支进行项目研发，严禁直接对 master 分支进行操作；
- 3) 每一个分支，经过内测、公测没问题的情况下，由项目经理合并到 master 分支上。

5.2.2 标签

- 1) 每一个大的研发节点完成后，项目经理需要基于当前分支或主分支进行“封板”

操作，即对当前项目完成情况做一个说明，同时对已上传的代码进行封存；

- 2) 基于标签，项目经理可以监控多个项目的进度，如：当前完成了那些功能；
- 3) 基于标签的描述，项目经理可以回滚项目代码到任意节点。