GIT

GIT

一、 什么是 GIT

Git 是一个开源的分布式版本控制系统，可以有效、高速的处理从很小到非常大的项目版本管理。*Git* 是 *Linus Torvalds* 为了帮助管理 *Linux* 内核开发而开发的一个开放源码的版本控制软件。

二、 GIT 的特点

分布式相比于集中式的最大区别在于开发者可以提交到本地，每个开发者通过克隆（*git clone*），在本地机器上拷贝一个完整的 *Git* 仓库。

从一般开发者的角度来看，*git* 有以下功能：

从服务器上克隆完整的 *Git* 仓库（包括代码和版本信息）到单机上、在自己的机器上根据不同的开发目的，创建分支，修改代码、在单机上自己创建的分支上提交代码、在单机上合并分支、把服务器上最新版的代码 *fetch* 下来，然后跟自己的主分支合并等。

优点：

适合分布式开发，强调个体。公共服务器压力和数据量都不会太大。速度快、灵活。任意两个开发者之间可以很容易的解决冲突。离线工作。

缺点：

资料少（起码中文资料很少）。学习周期相对而言比较长。不符合常规思维。代码保密性差，一旦开发者把整个库克隆下来就可以完全公开所有代码和版本信息。

因其资料的公开性，导致大型商业化工程几乎不会使用 *GIT* 来托管工程版本信息（除非搭建企业私服）。

三、 Github 简介

平台地址：*https://github.com*

gitHub 是一个面向开源及私有软件项目的托管平台，因为只支持 *git* 作为唯一的版本库格式进行托管，故名 *gitHub*。

gitHub 于 *2008* 年 *4* 月 *10* 日正式上线，除了 *git* 代码仓库托管及基本的 *Web* 管理界面以外，还提供了订阅、讨论组、文本渲染、在线文件编辑器、协作图谱（报表）、代码片段分享（*Gist*）等功能。目前，其注册用户已经超过 *350* 万，托管版本数量也是非常之多，其中不乏知名开源项目 *Rubyon Rails*、*jQuery*、*python* 等。

作为开源代码库以及版本控制系统，*Github* 拥有超过 *900* 万开发者用户。随着越来越多的应用程序转移到了云上，*Github* 已经成为了管理软件开发以及发现已有代码的首选方法。

如前所述，作为一个分布式的版本控制系统，在 *Git* 中并不存在主库这样的概念，每一份复制出的库都可以独立使用，任何两个库之间的不一致之处都可以进行合并。

在 *GitHub*，用户可以十分轻易地找到海量的开源代码。

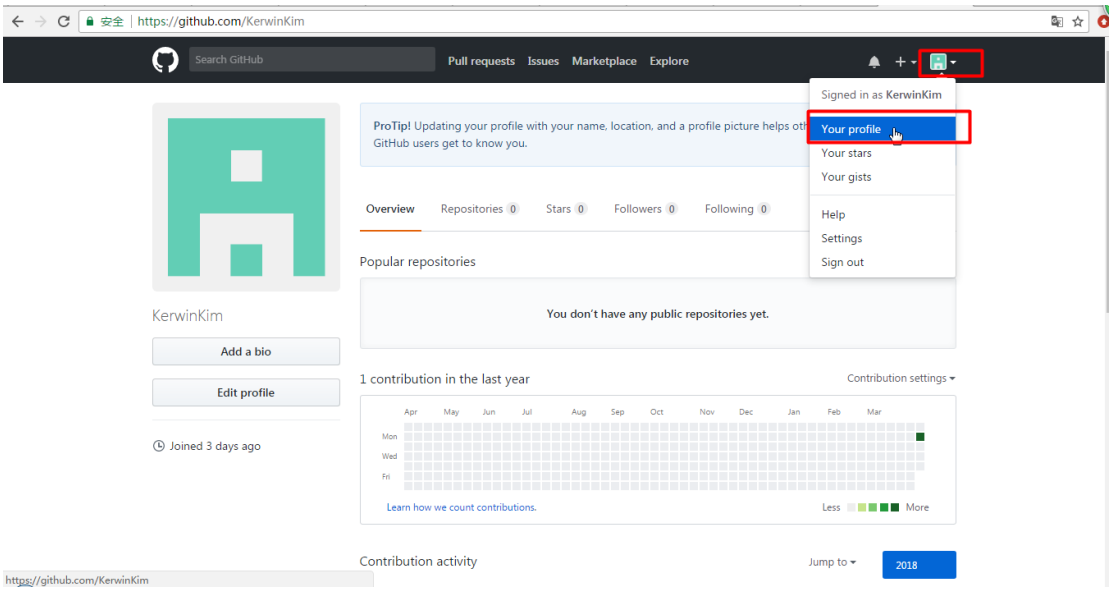
1 注册账号

请自行在 *Github*（*https://github.com*）网站中注册。

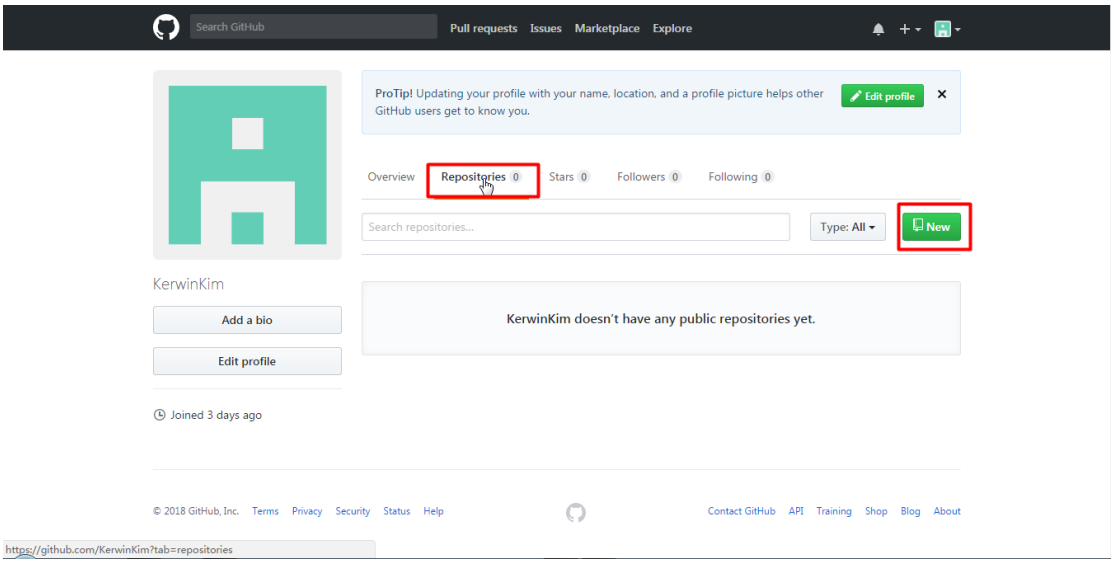
在部分企业中，开发人员是否拥有 *Github* 帐户，在 *Github* 中是否有个人的代码和资料发布、发布数量等，成为了开发人员实力的一种评价标准。

2 创建版本仓库

2.1 进入管理页面

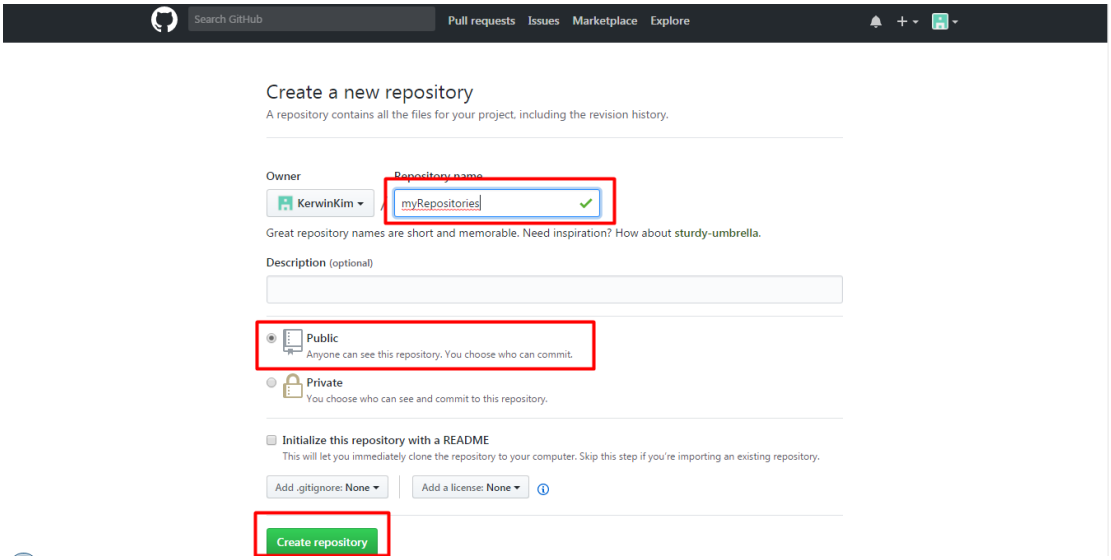


2.2 进入仓库管理面板

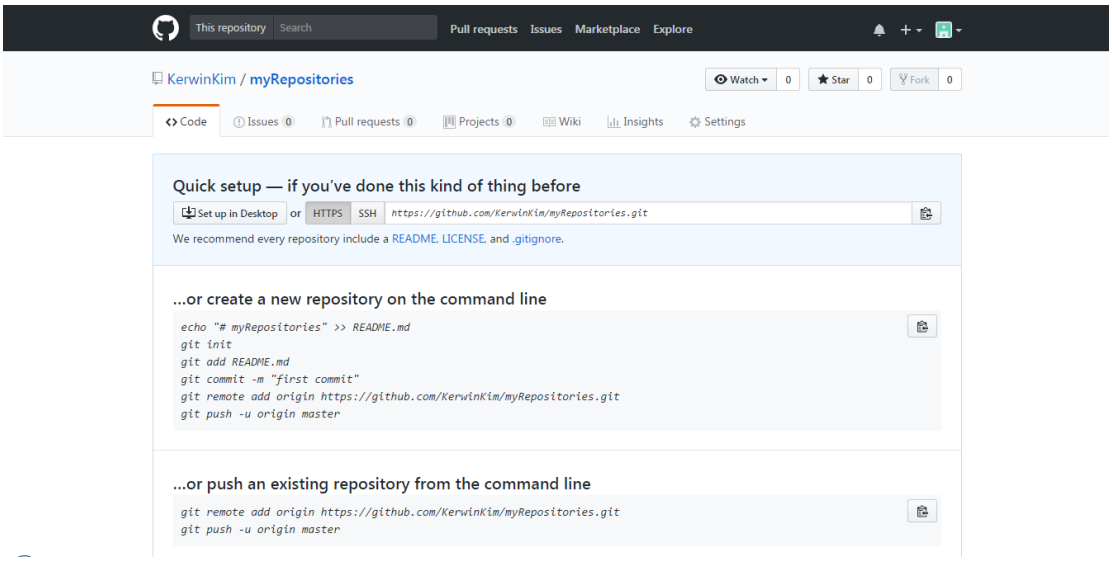


2.3 提供新增仓库信息

注意，这里创建的是公开版本仓库，私有版本仓库不完全开放，大部分功能需要付费购买。



2.4 成功创建后的页面内容

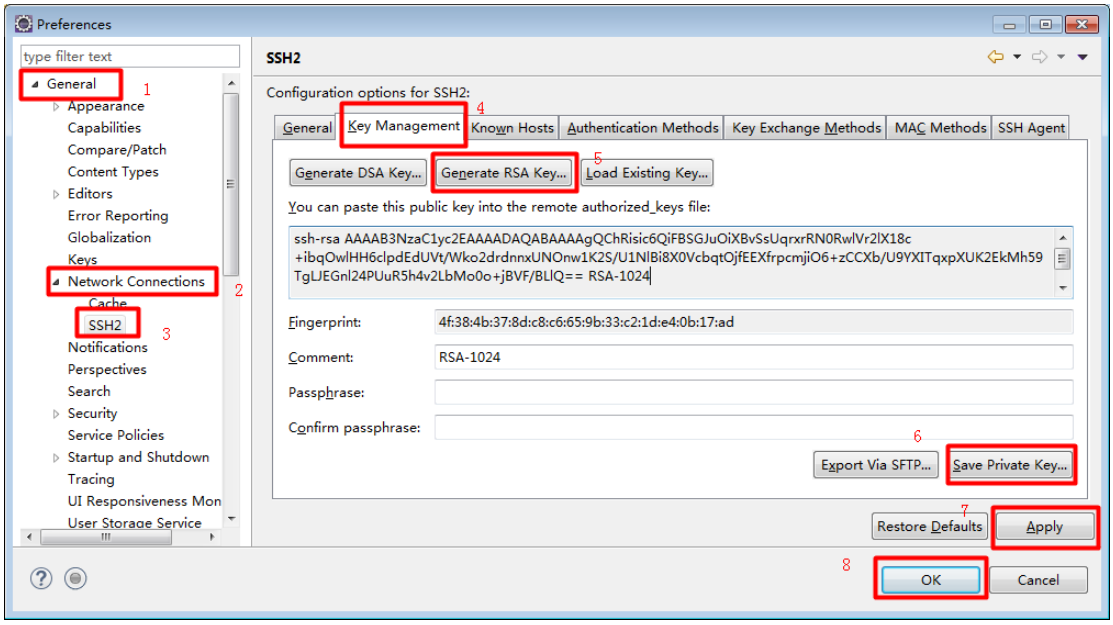


3 提供 SSH2 密匙

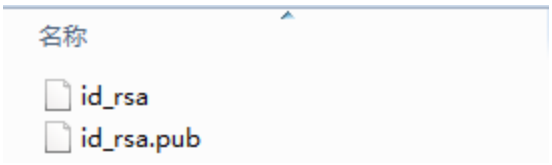
3.1 创建本地密匙

使用 *EclipseIDE* 生成本地密匙文件。

*window->preference->general->network connections->ssh2->*点击 *Generate RSA KEY->*点击 *Save private key* 生成并保存本地密匙：

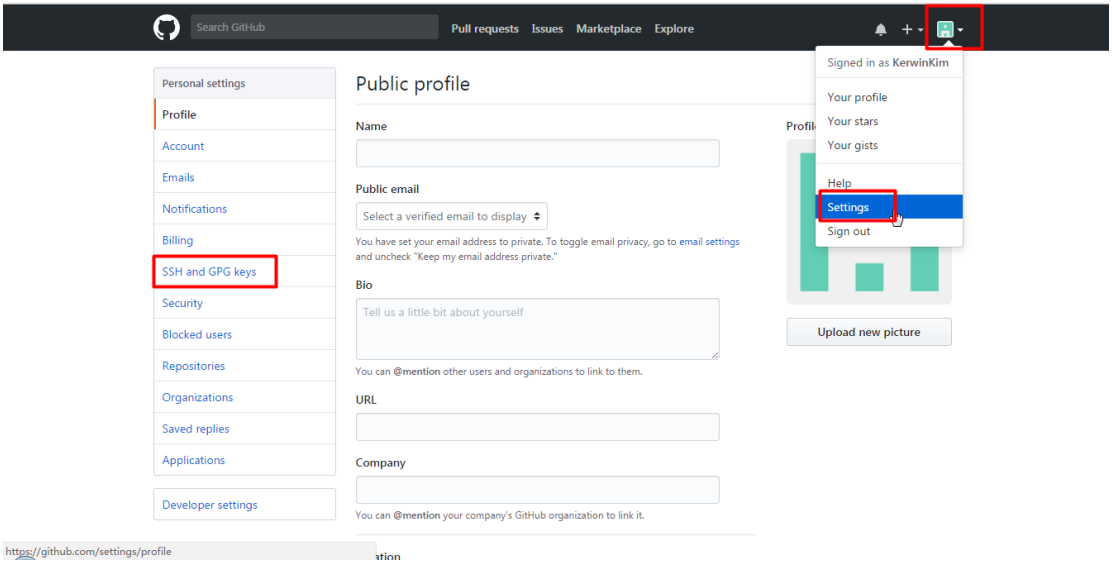


生成后的密匙文件所在位置是：*C:\${user.home}\.ssh* 目录。密匙文件名为：*id_rsa* 和 *id_rsa.pub*。其中 *id_rsa.pub* 文件是公钥密匙，需要手工添加到 *Github* 密匙库中。

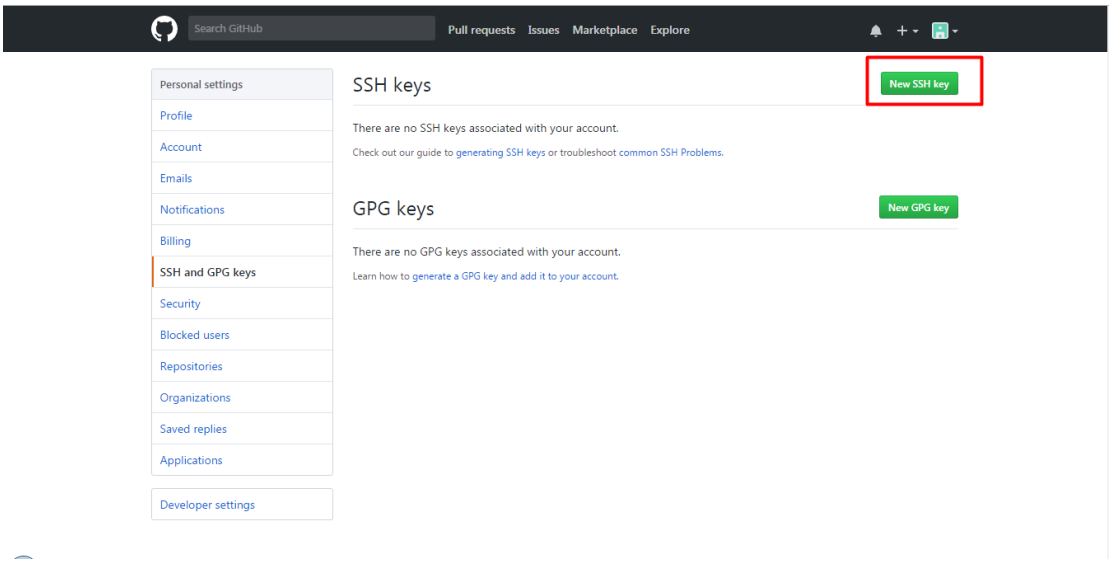


3.2 Github 添加密匙

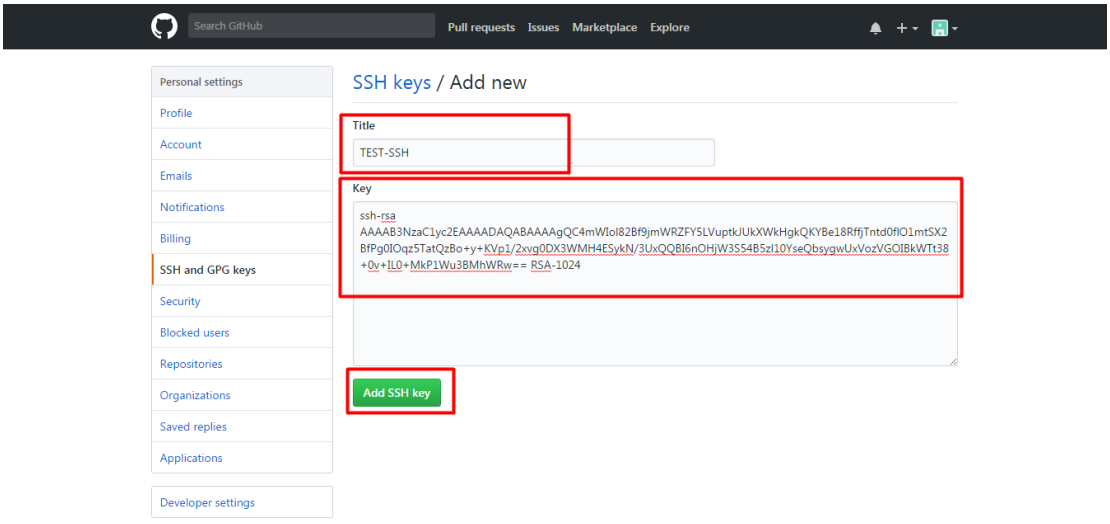
进入 *Github* 密匙管理面板



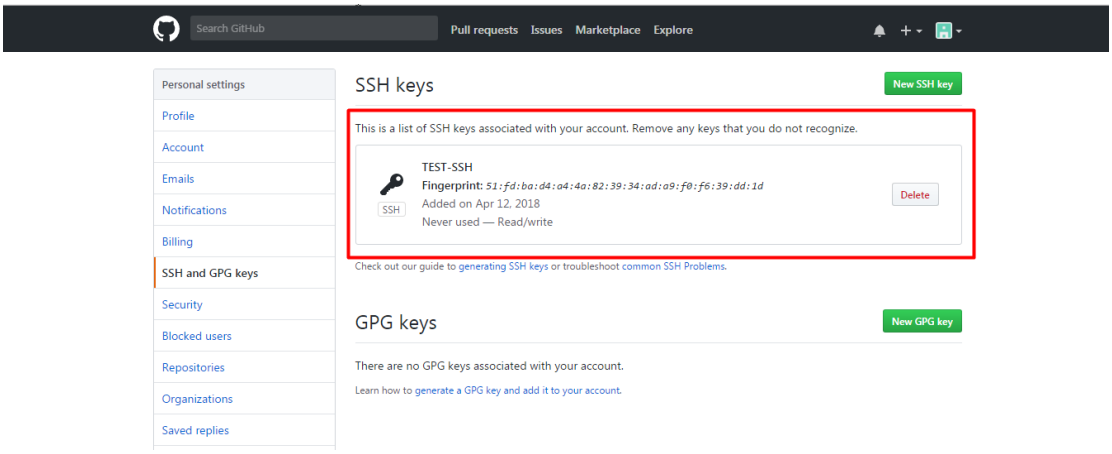
创建新的 *SSH* 密匙：



在 *title* 中输入密匙名称（自定义），在 *key* 文本域中输入 *id_rsa.pub* 文件中的内容。并确认新增密匙。

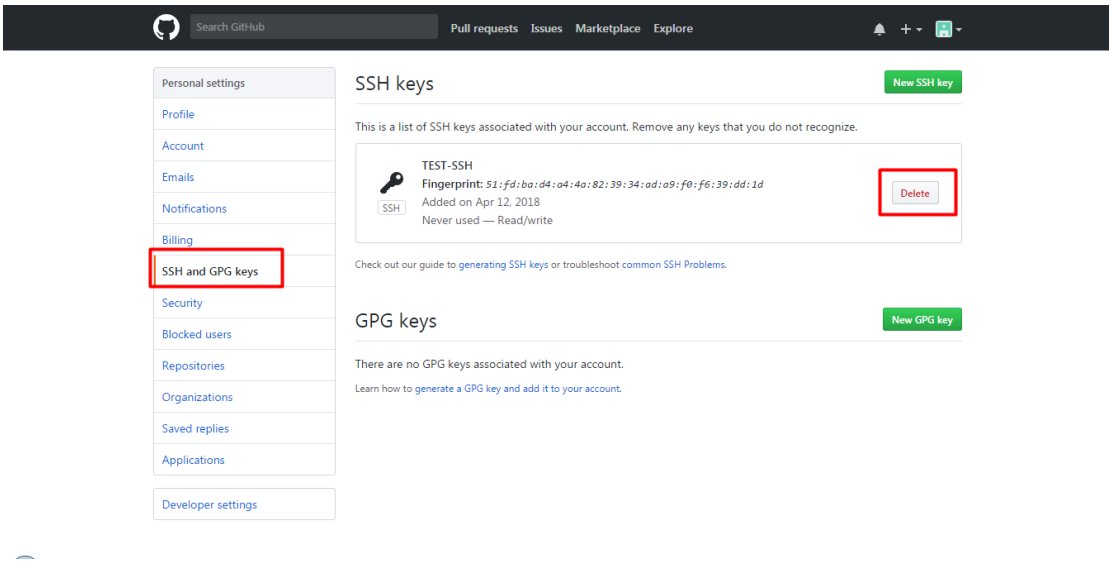


密匙新增成功（新增密匙过程可能需要确认用户密码），管理面板如下：



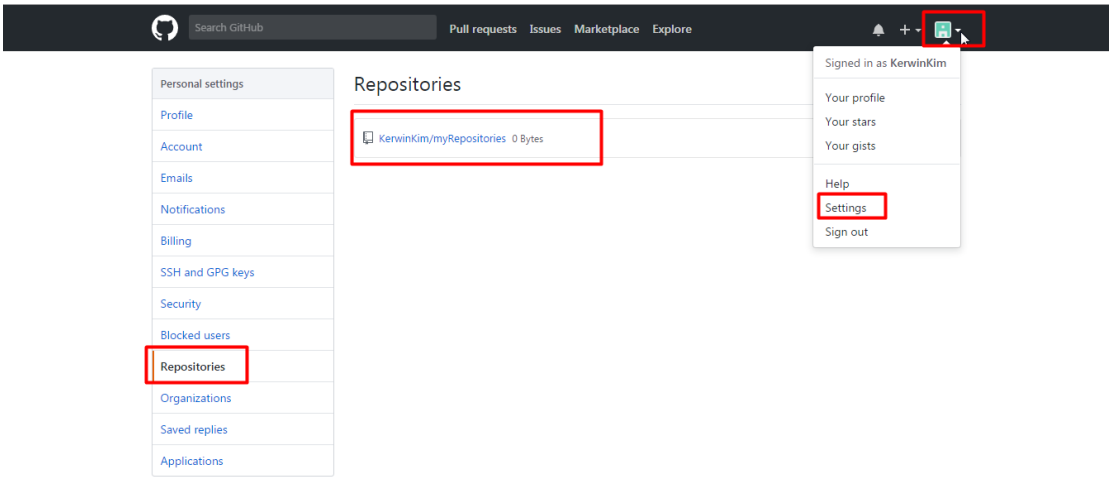
3.3 Github 删除密匙

在密匙管理面板中，点击 *Delete* 按钮可以删除密匙（删除过程可能需要输入登录密码）。

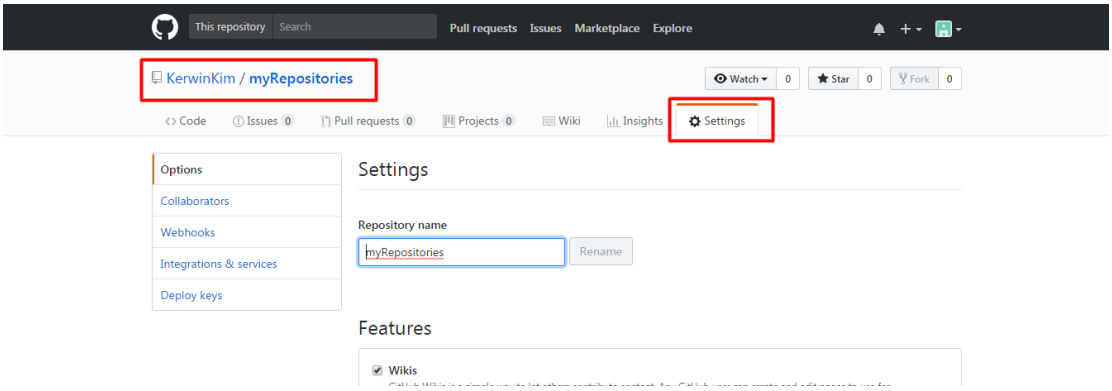


4 删除版本仓库

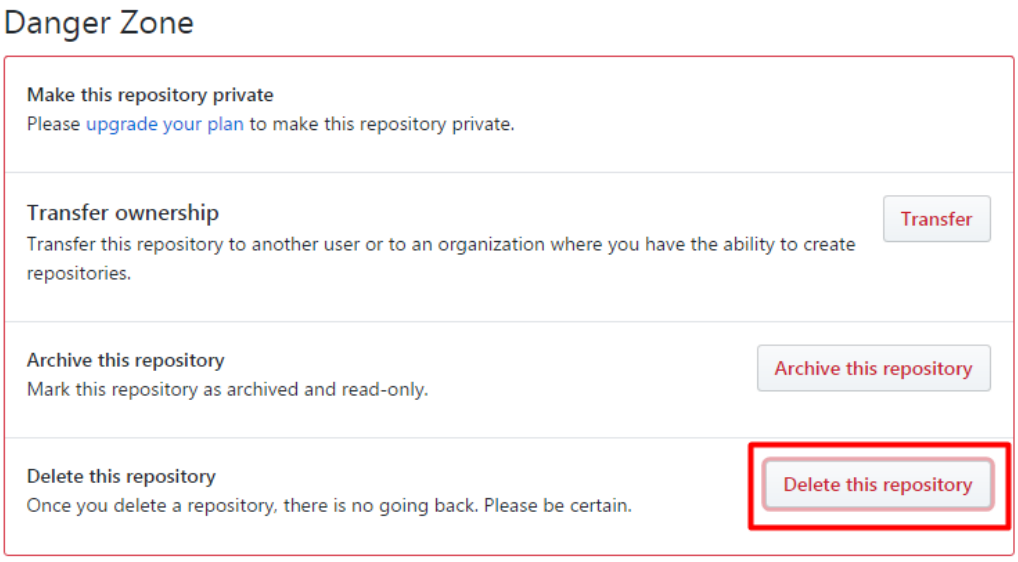
进入版本库管理面板：



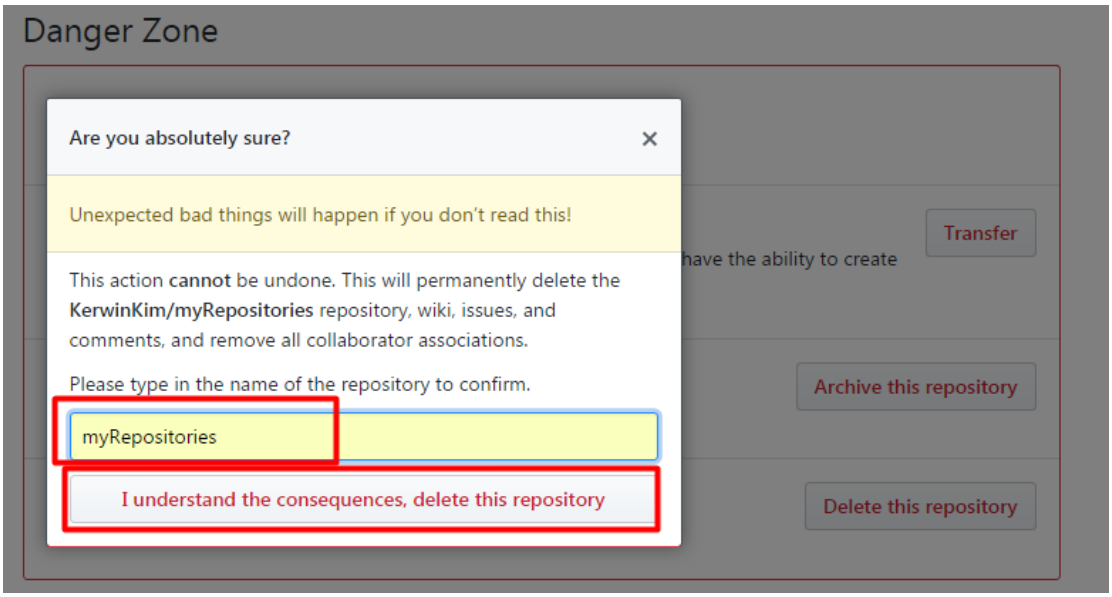
选择要删除的版本仓库，进入对应仓库的 *Settings* 管理界面：



在管理界面的最末端，点击 *Delete this repository* 按钮：



在确认对话框中，输入要删除的版本仓库名称，并确认删除：

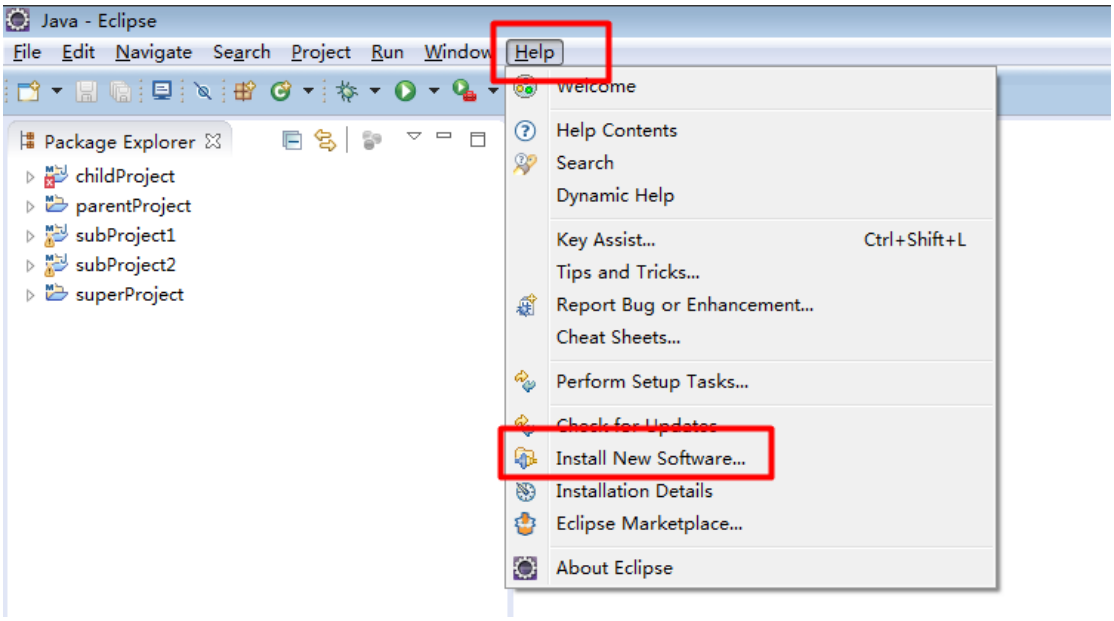


四、 EGIT 插件应用（Eclipse GIT 插件）

1 安装 EGIT 插件

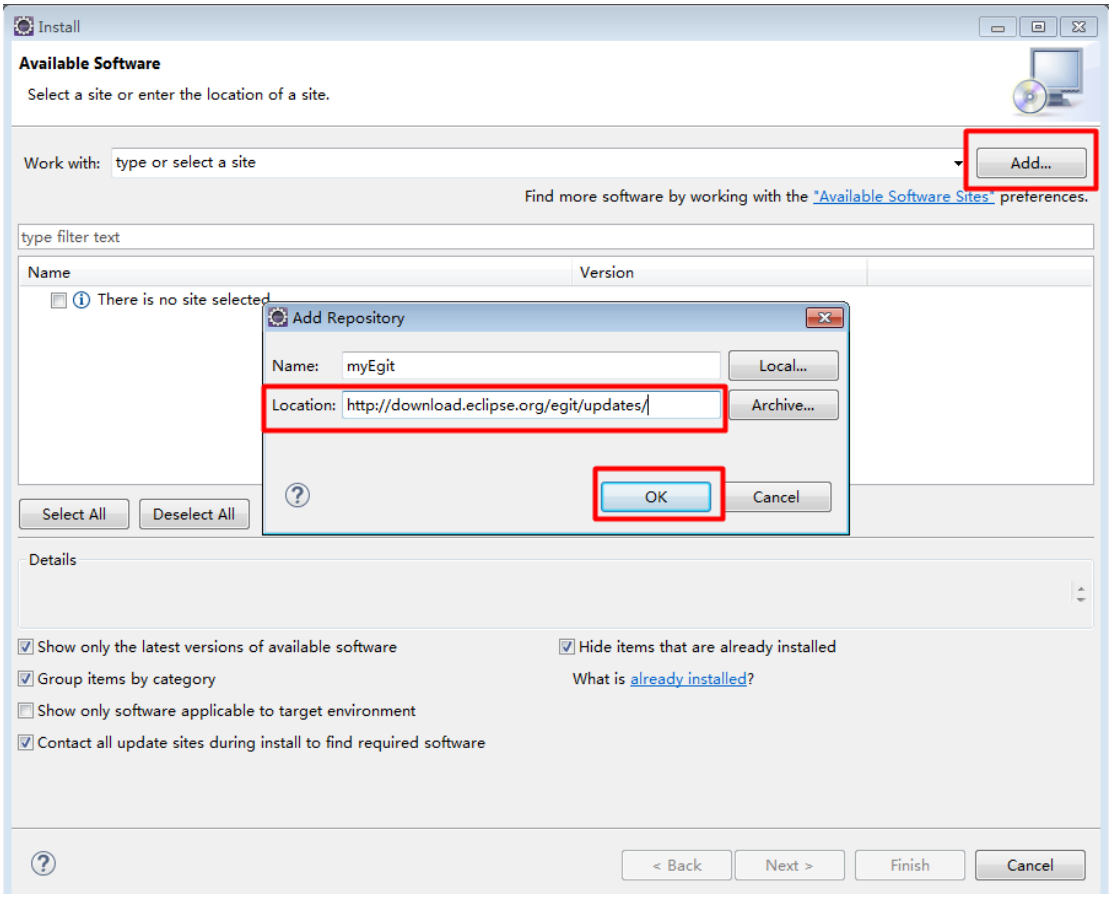
Eclipse Mars2 版本，默认集成 *EGIT* 插件，如果需要安装 *EGIT* 插件，可以使用在线安装方式实现。

1.1 进入新插件安装面板：

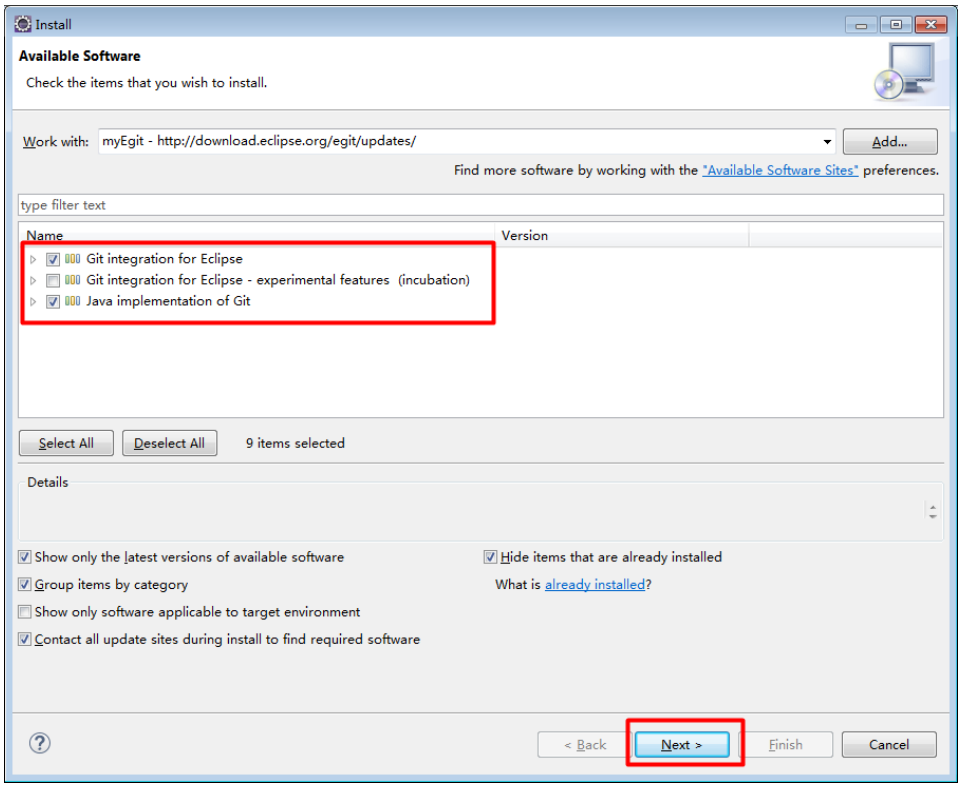


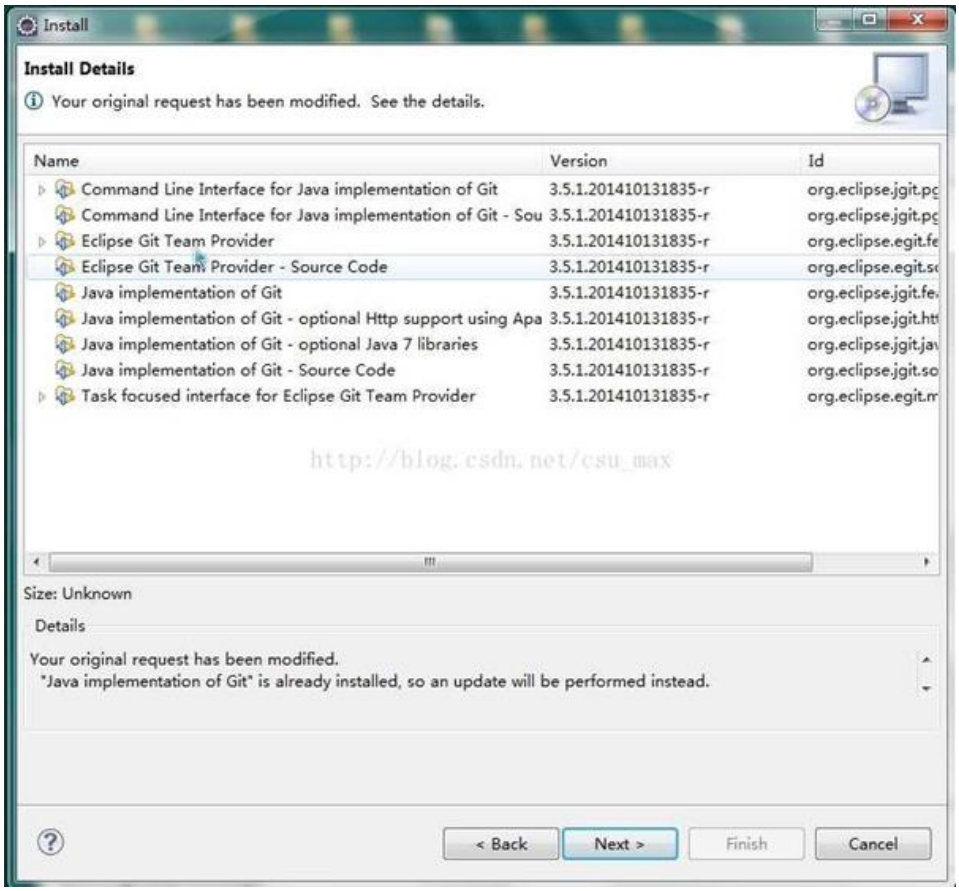
1.2 新增要安装的插件

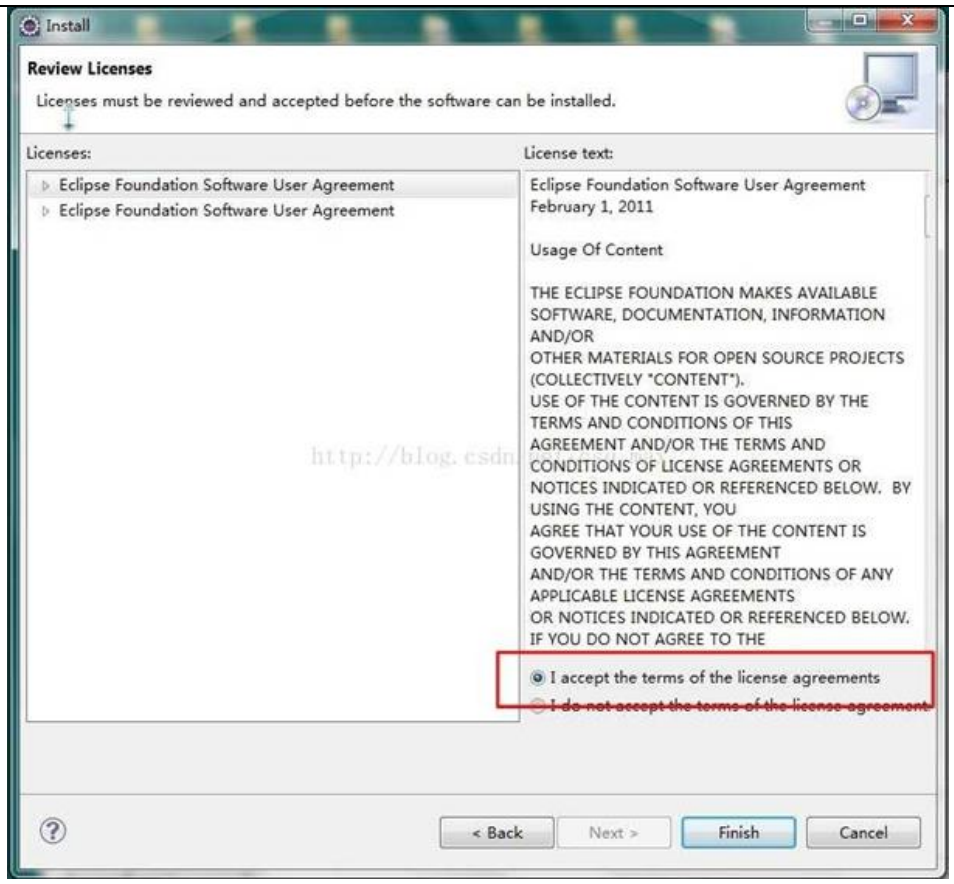
插件名称自定义， 插件地址为：http://download.eclipse.org/egit/updates/。



1.3 选择要拉取的插件







2 Eclipse 访问 Github 问题解决

ini 配置文件增加下述配置：

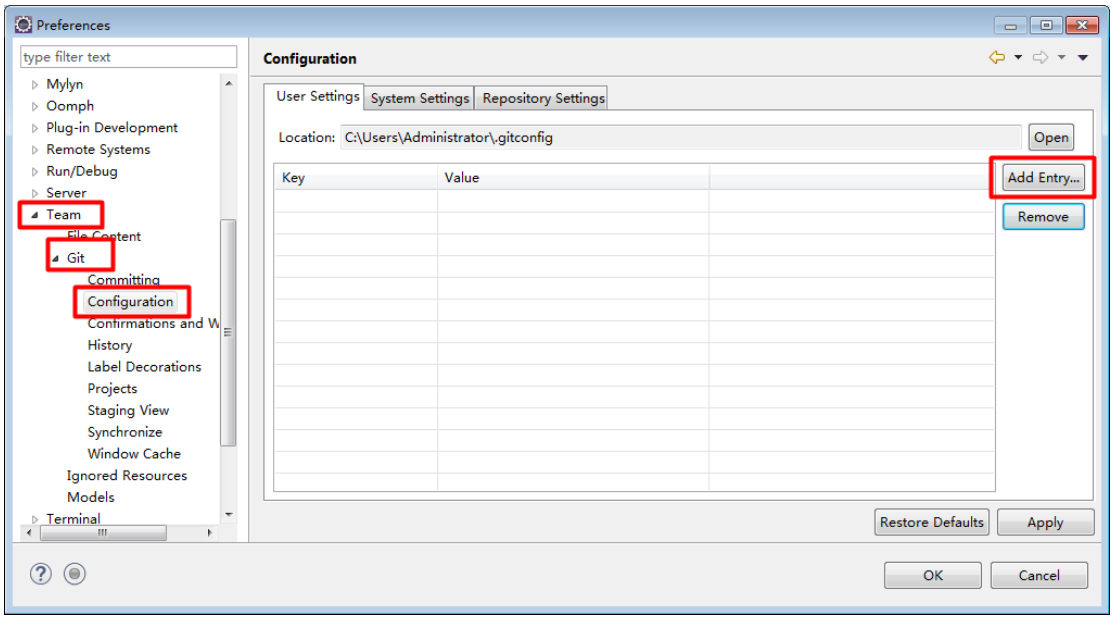
ini 配置文件位置： *eclipse* 安装目录*/eclipse.ini* 文件。

```
-Dhttps.protocols=TLSv1.1,TLSv1.2
```

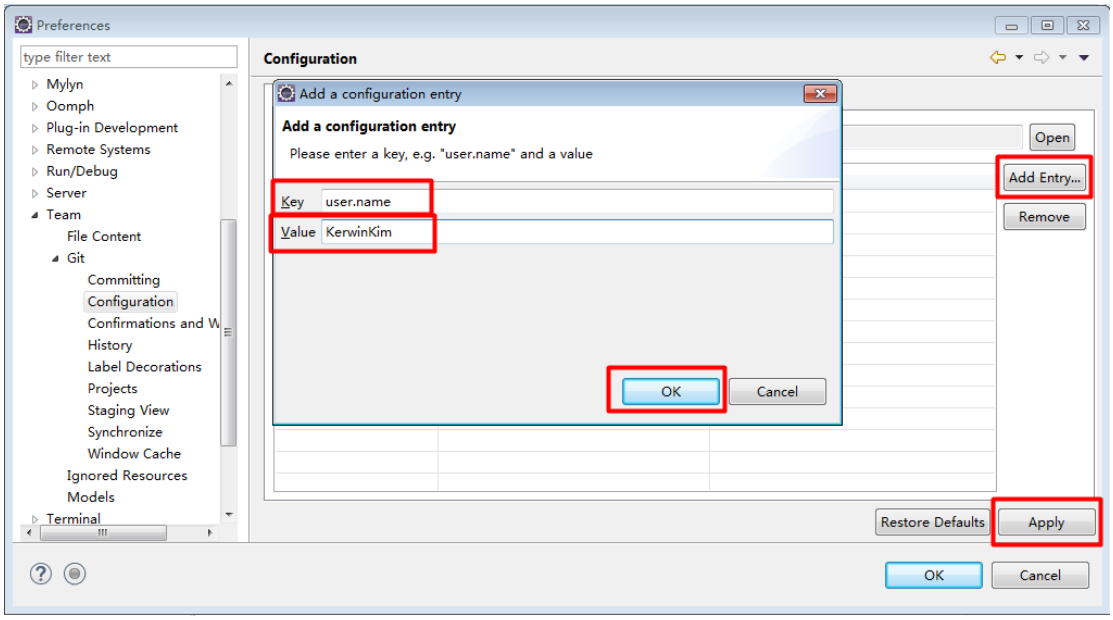
3 EGIT 插件参数配置

使用 *EGIT* 插件，可以提前配置一些参数，为 *GIT* 访问远程版本仓库提供便利。

3.1 进入 Eclipse 配置面板（Window->Preferences）。



3.2 新增参数信息



常用参数有下述 4 个：

http.sslVerify=false 关闭 *ssl* 校验。***（选填，如果未提供 SSH2 密匙必填）***

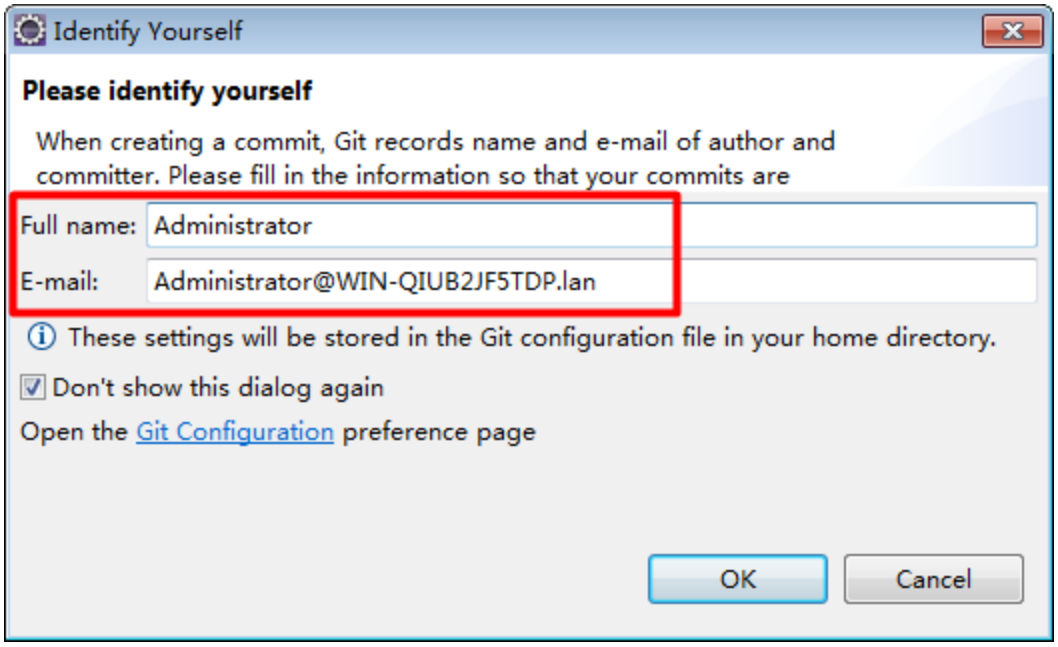
http.sslVersion=tlsv1.2 定义 *ssl* 协议版本，*Github* 在最近的更新中关闭了 *tlsv1.0* 和 *tlsv1.1* 协议的访问，不提供此参数无法访问 *Github* 远程版本仓库 ***（JDK1.8 未测试）***。

user.email=xxx@xxx

user.name=xxx

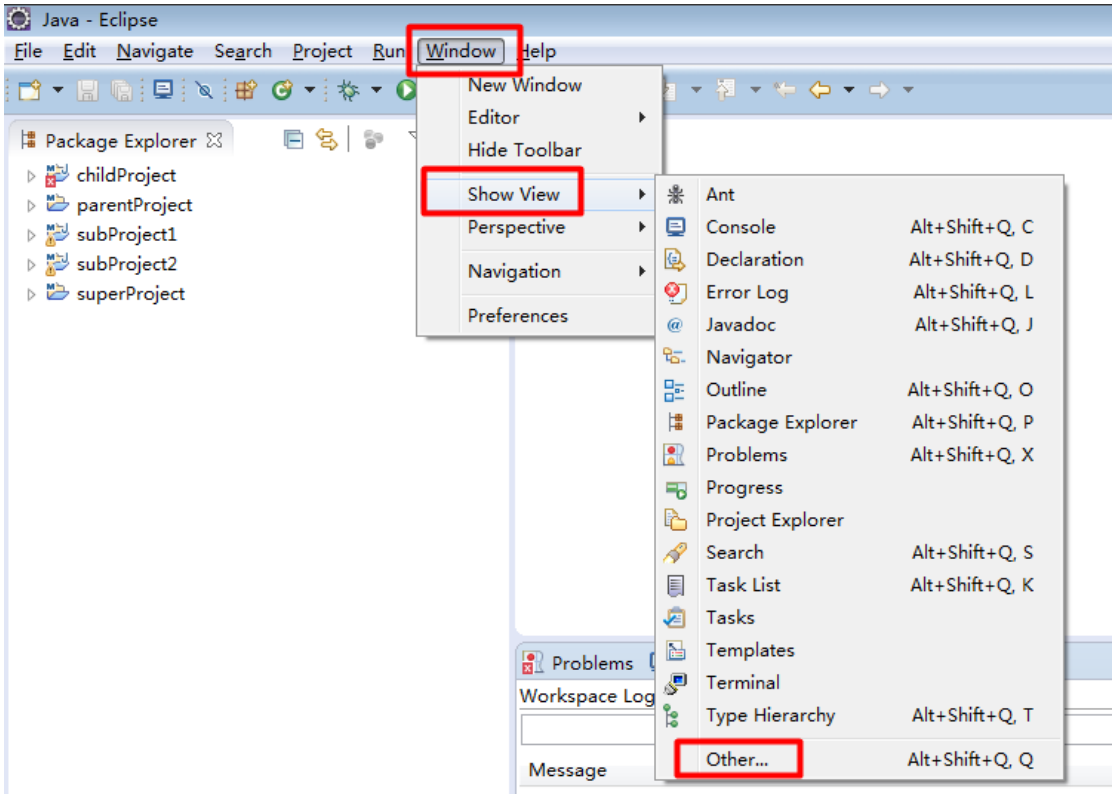
在访问 *Github* 远程版本仓库的时候，*Github* 要求必须提供用户和电子邮箱，如果不提

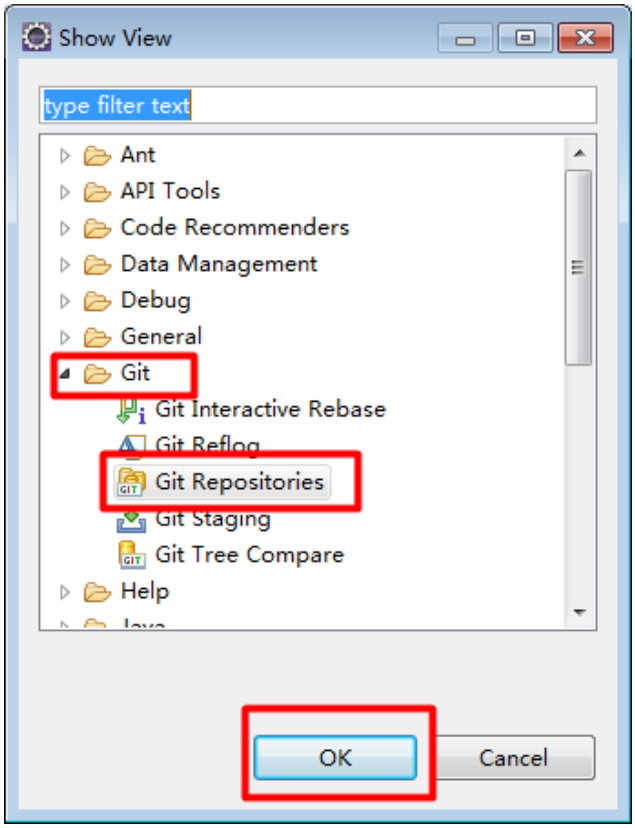
供上述参数，在后续操作中需要提供对应的信息，参考下图。

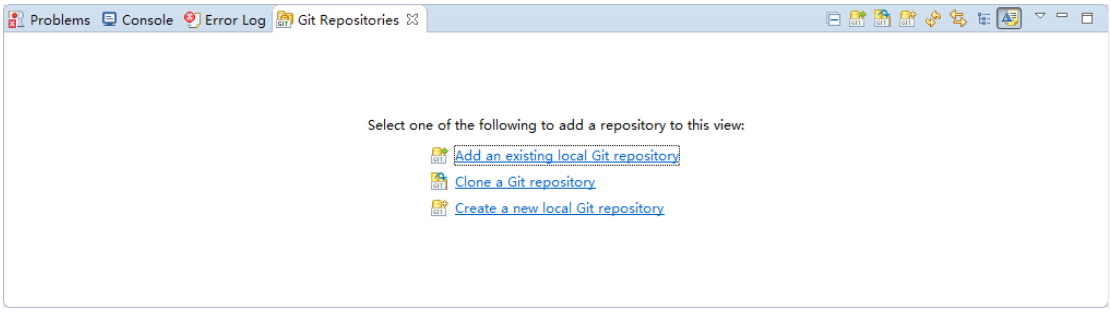


4 创建本地版本仓库

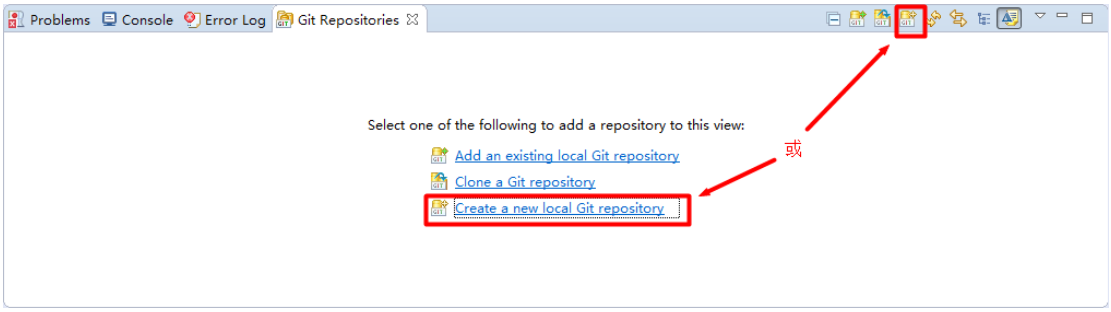
4.1 打开 GIT Repositories 管理面板

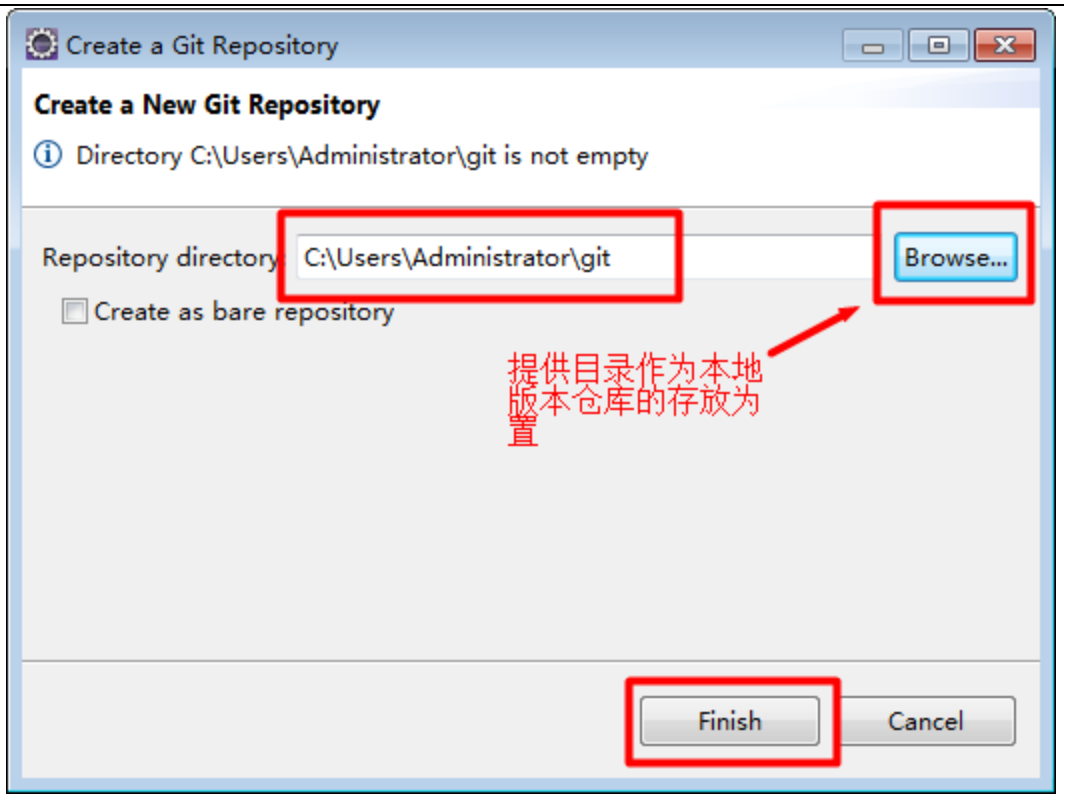


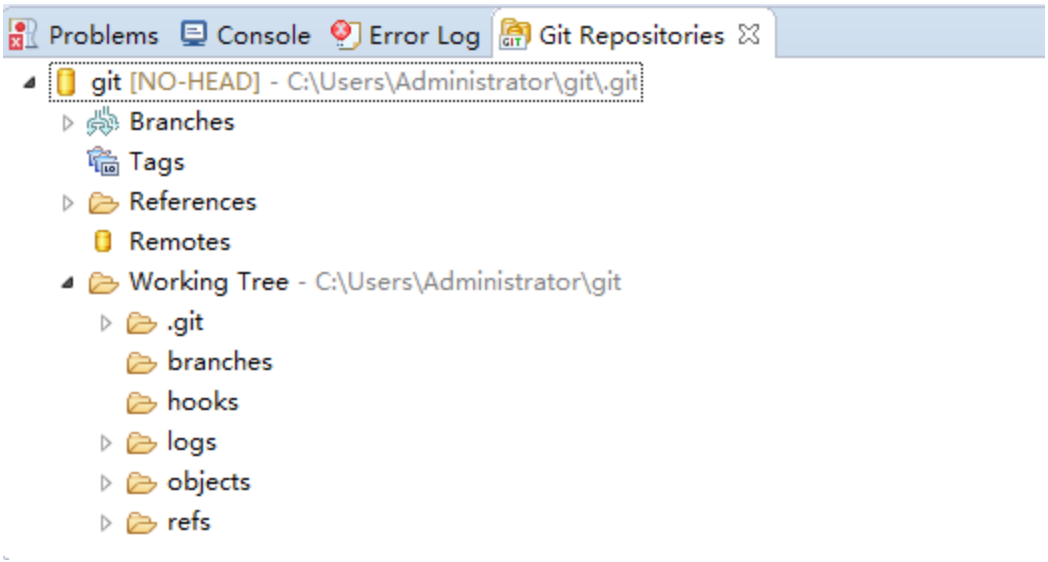




4.2 创建本地版本仓库

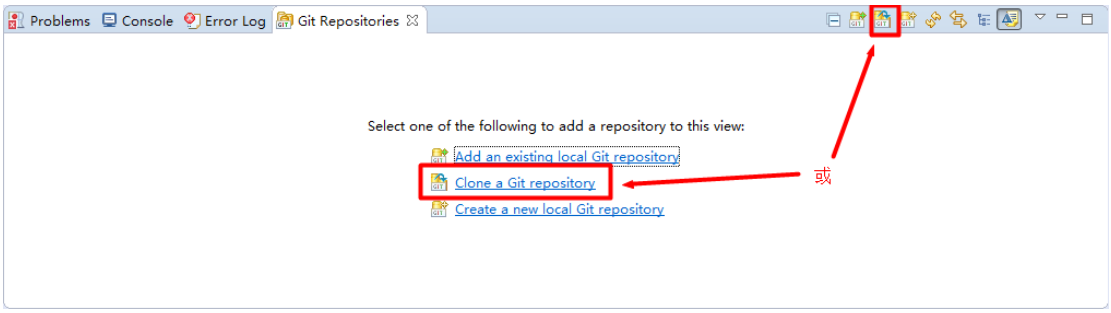


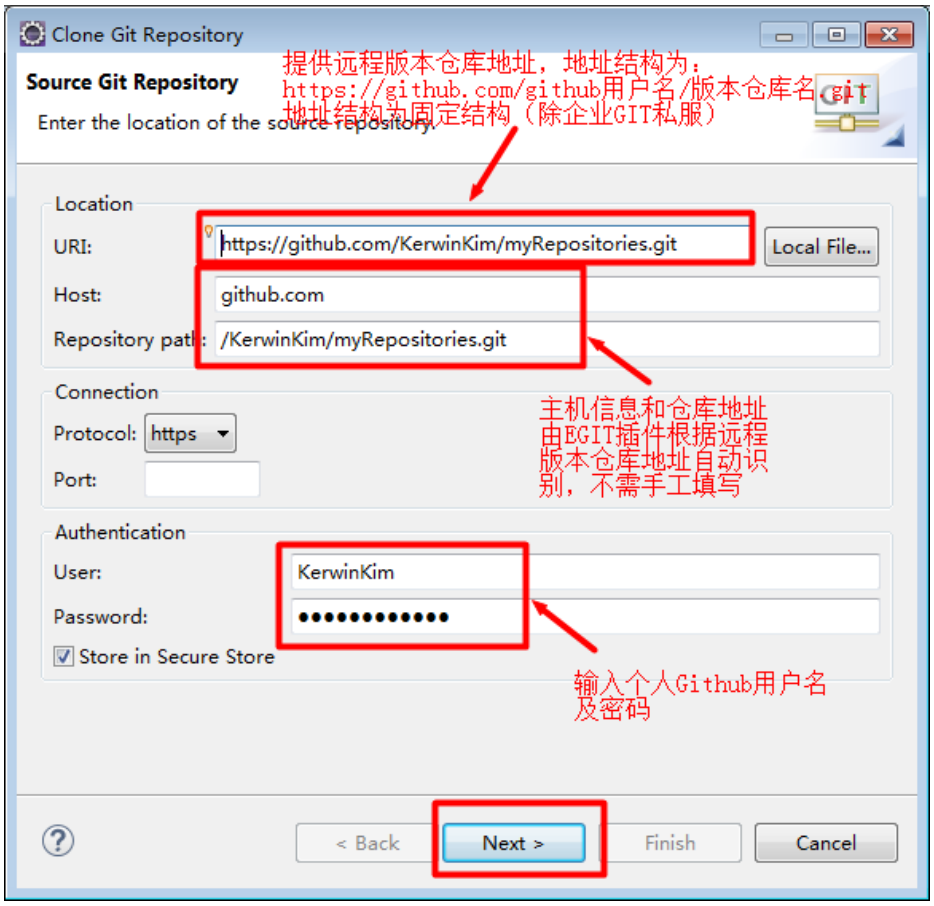


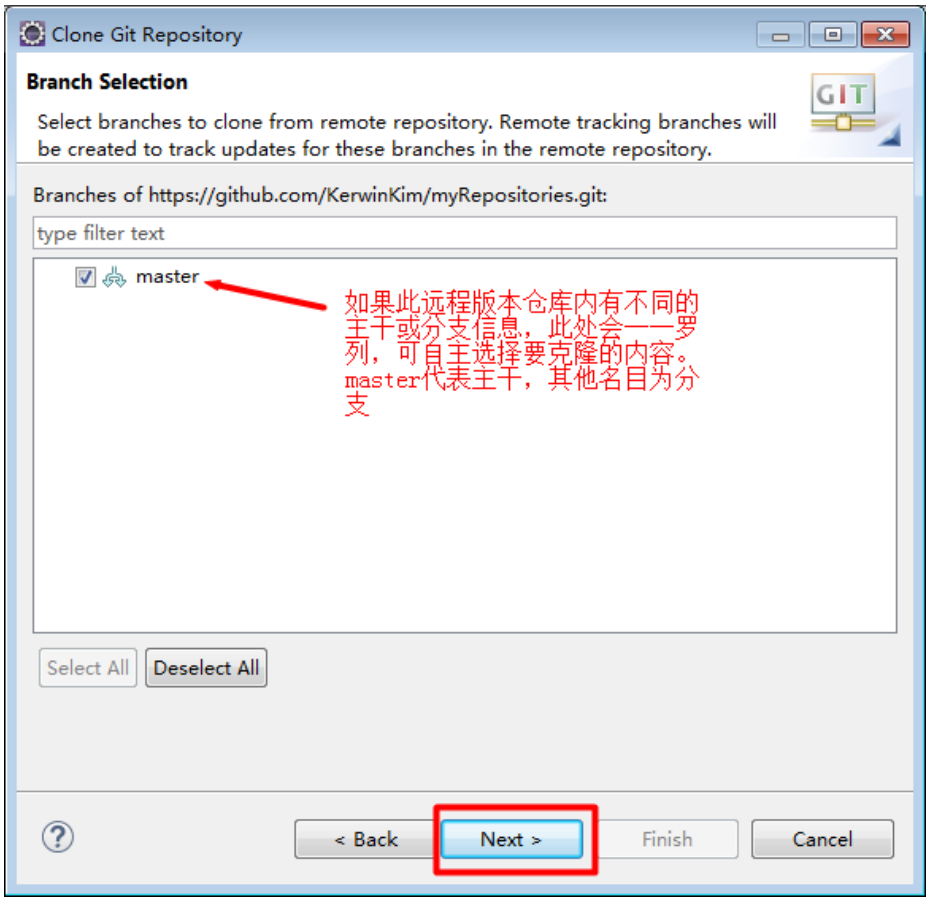


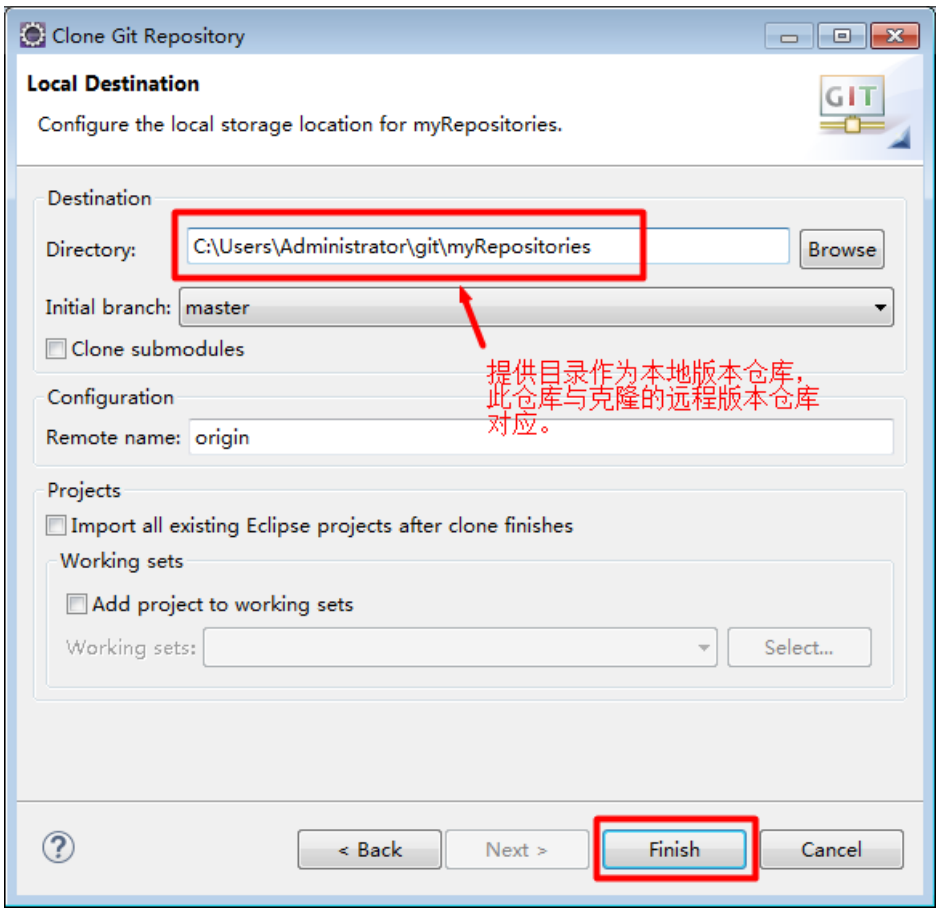
5 克隆远程版本仓库

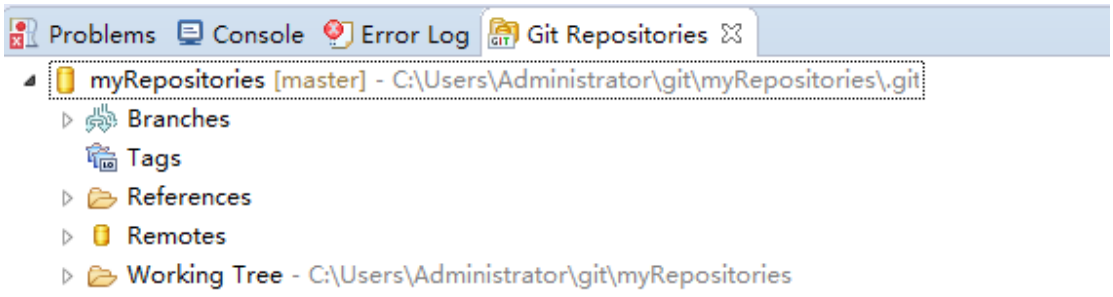
此操作为常用操作，在克隆远程版本仓库同时，EGIT 插件会自动创建一个本地版本仓库与之对应。


Problems
Console
Error Log
Git Repositories
Select one of the following to add a repository to this view:
Add an existing local Git repository
Clone a Git repository
Create a new local Git repository
或


Clone Git Repository
Source Git Repository
Enter the location of the source repository.
提供远程版本仓库地址，地址结构为：
https://github.com/github用户名/版本仓库名
地址结构为固定结构（除企业GIT私服）
Location
URI:
https://github.com/KerwinKim/myRepositories.git
Local File...
Host:
github.com
Repository path:
/KerwinKim/myRepositories.git
Connection
Protocol:
https
Port:
主机信息和仓库地址
由EGIT插件根据远程
版本仓库地址自动识
别，不需手工填写
Authentication
User:
KerwinKim
Password:
Store in Secure Store
输入个人Github用户名
及密码
< Back
Next >
Finish
Cancel

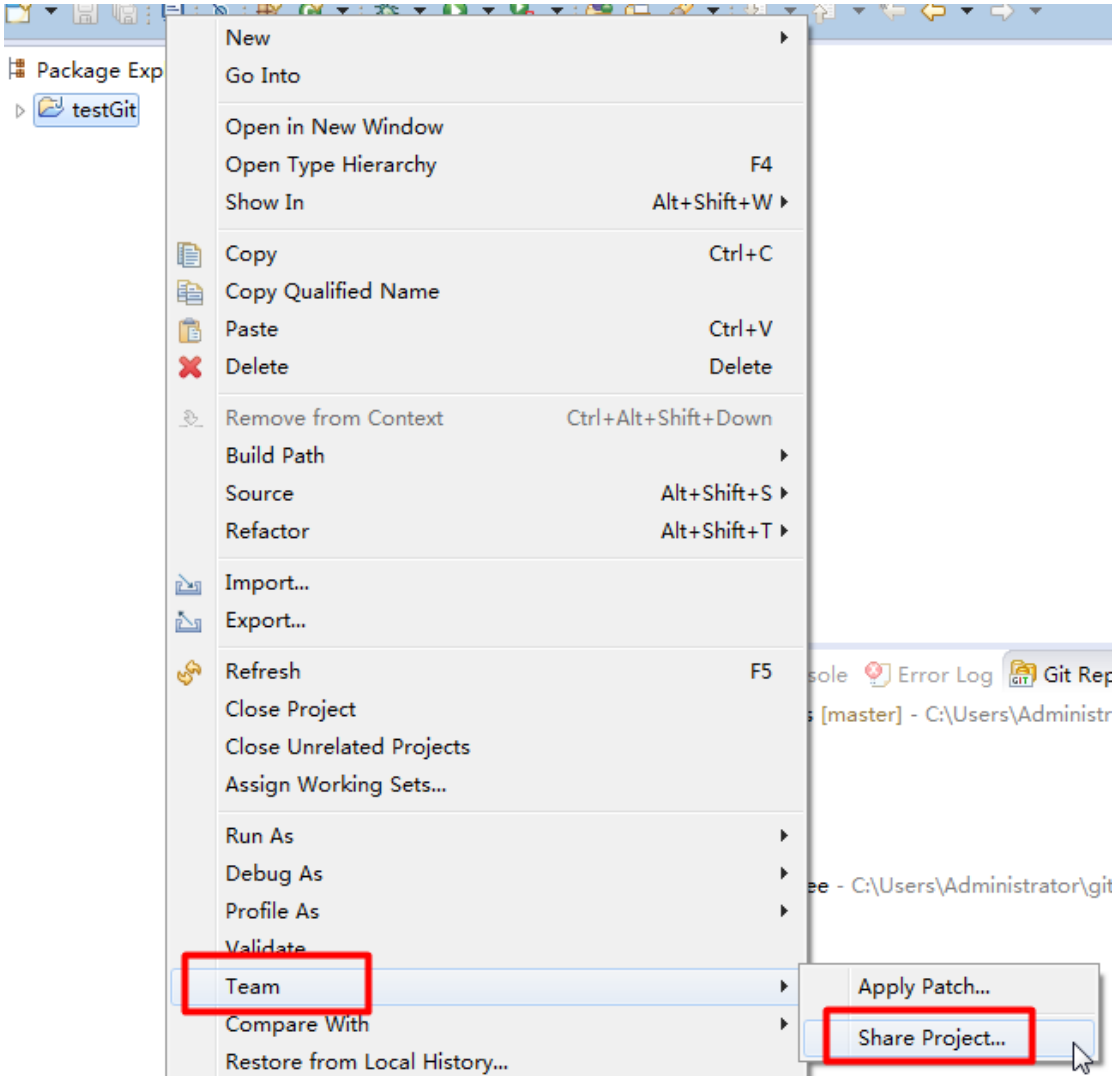

Clone Git Repository
Branch Selection
Select branches to clone from remote repository. Remote tracking branches will be created to track updates for these branches in the remote repository.
Branches of https://github.com/KerwinKim/myRepositories.git:
type filter text
master
如果此远程版本仓库内有不同的主干或分支信息，此处会一一罗列，可自主选择要克隆的内容。master代表主干，其他名目为分支
Select All
Deselect All
< Back
Next >
Finish
Cancel

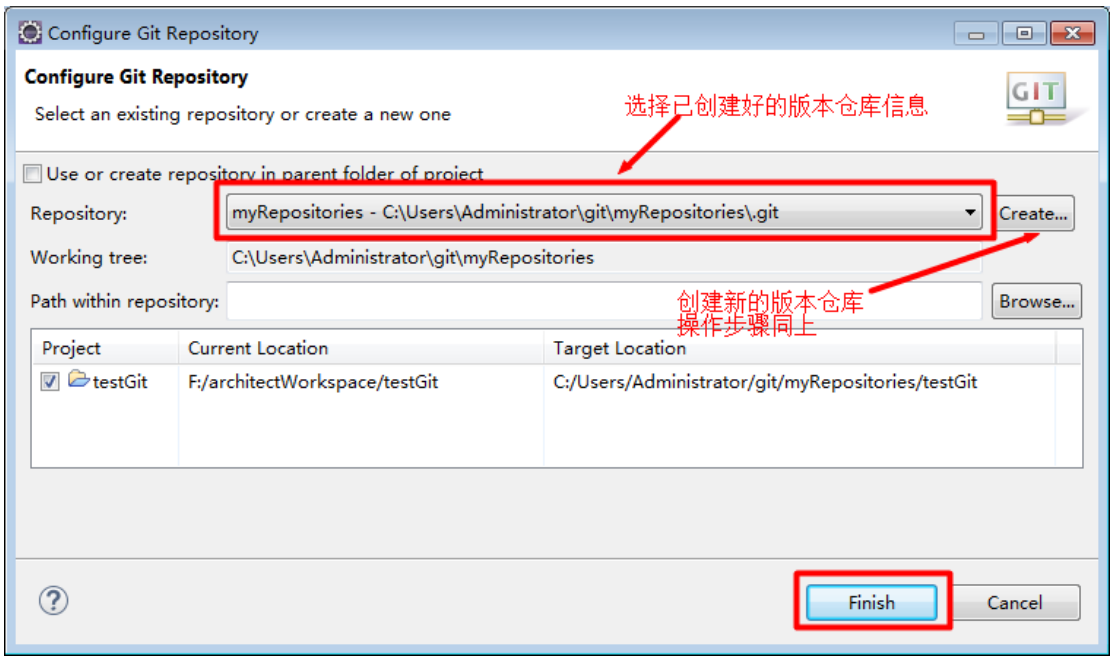

Clone Git Repository
Local Destination
Configure the local storage location for myRepositories.
Destination
Directory:
C:\Users\Administrator\git\myRepositories
Browse
Initial branch:
master
Clone submodules
提供目录作为本地版本仓库，此仓库与克隆的远程版本仓库对应。
Configuration
Remote name:
origin
Projects
Import all existing Eclipse projects after clone finishes
Working sets
Add project to working sets
Working sets:
Select...
< Back
Next >
Finish
Cancel

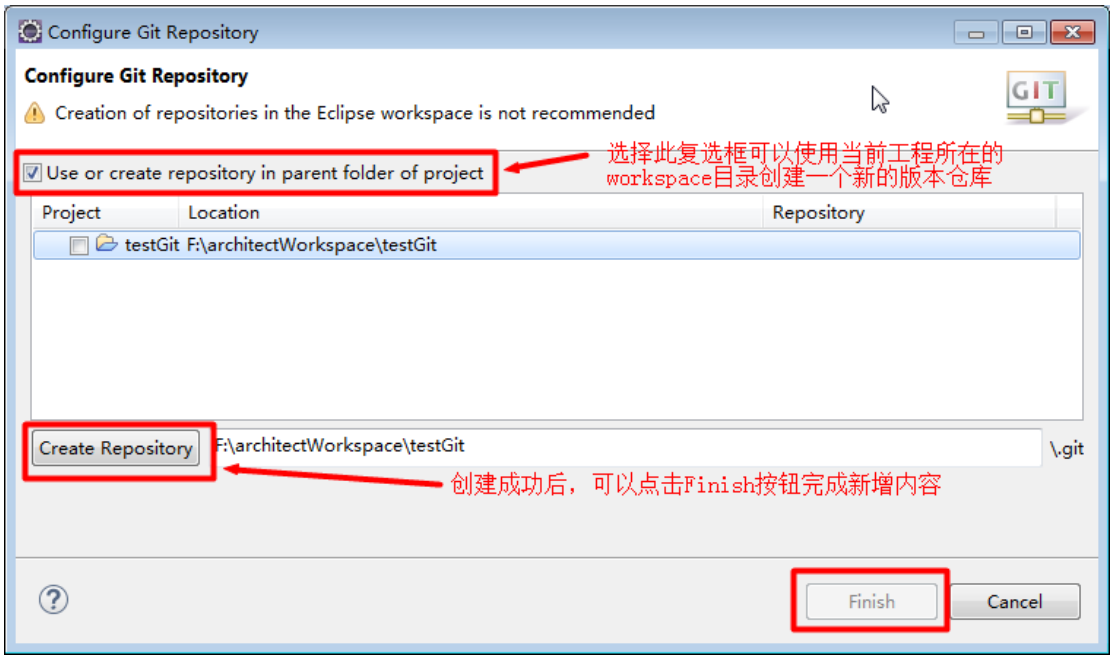


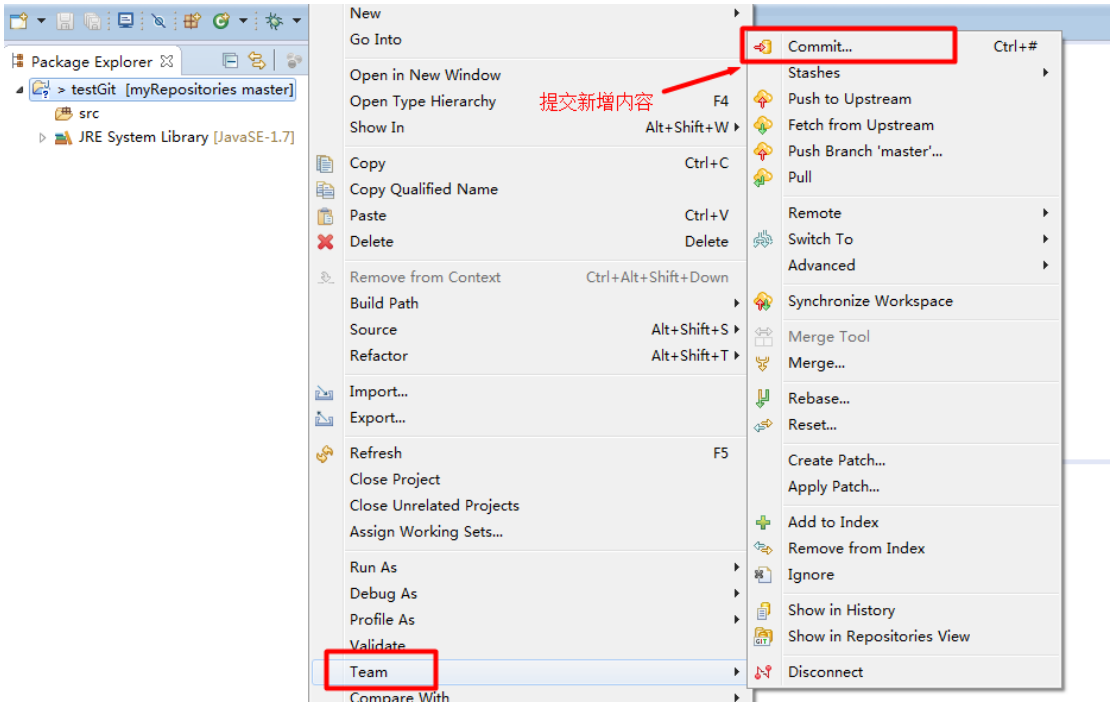
6 增加内容

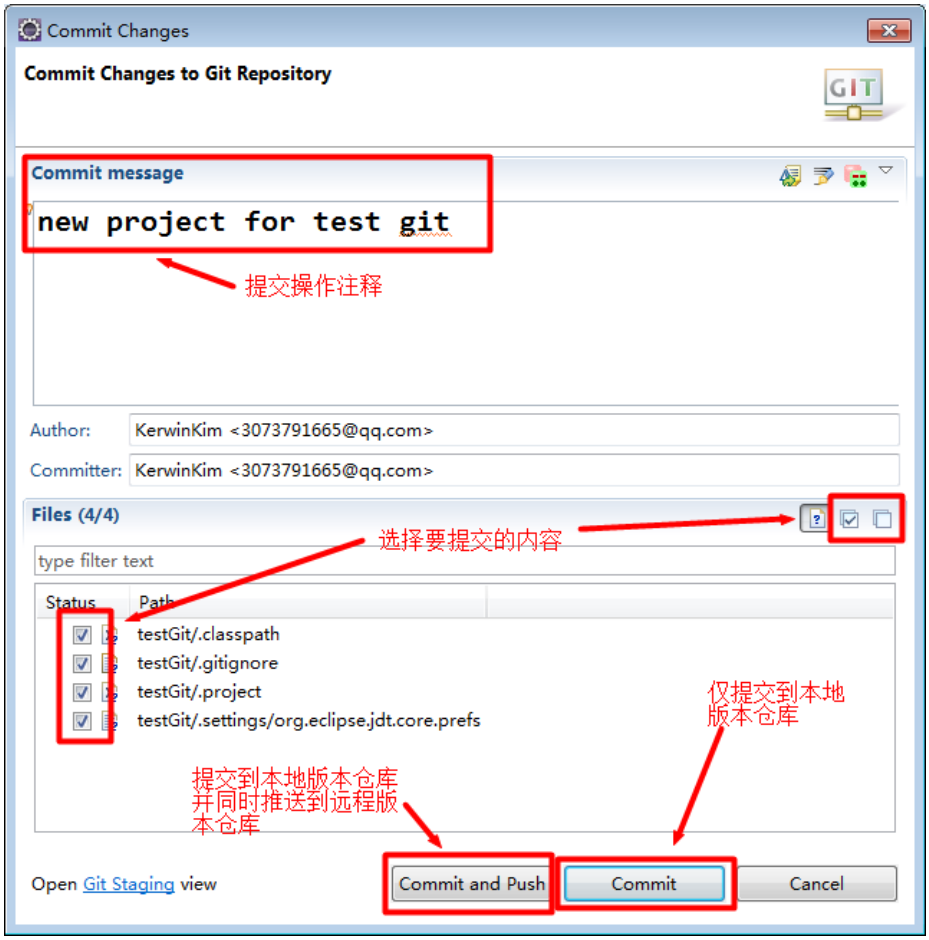
此操作就是将本地工程分享到 *GIT* 中，并使用 *GIT* 来进行版本信息管理。

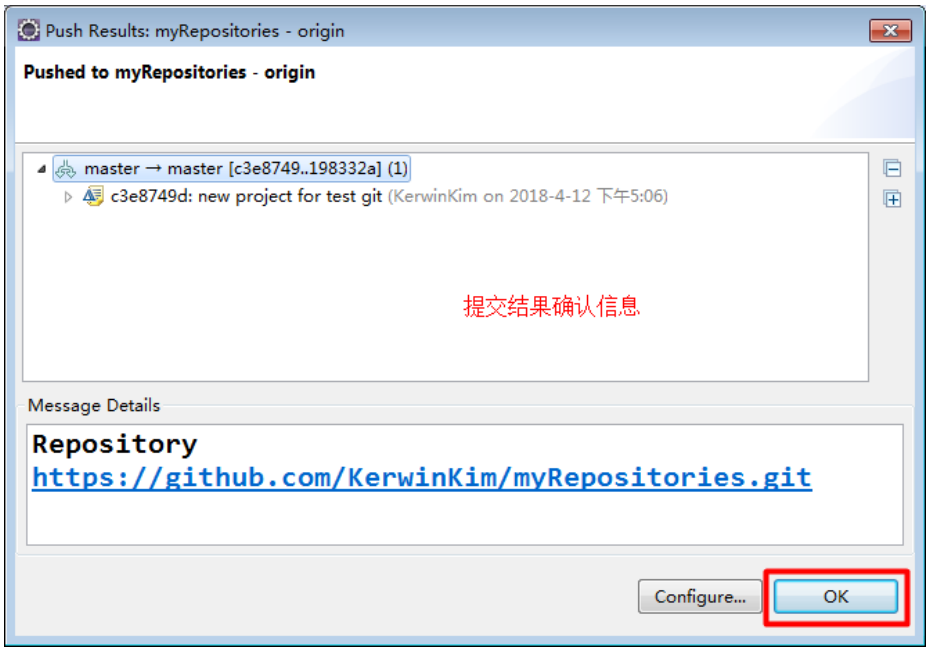


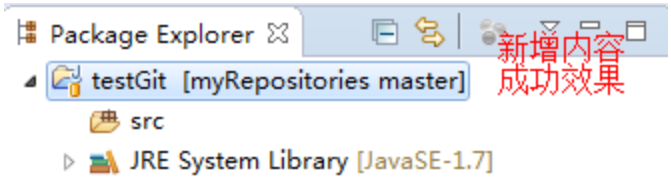

Configure Git Repository
Configure Git Repository
Select an existing repository or create a new one
选择已创建好的版本仓库信息
Use or create repository in parent folder of project
Repository:
myRepositories - C:\Users\Administrator\git\myRepositories\.git
Create...
Working tree:
C:\Users\Administrator\git\myRepositories
Path within repository:
Browse...
创建新的版本仓库
操作步骤同上
Project
Current Location
Target Location
testGit
F:/architectWorkspace/testGit
C:/Users/Administrator/git/myRepositories/testGit
Finish
Cancel


Configure Git Repository
Configure Git Repository
Creation of repositories in the Eclipse workspace is not recommended
Use or create repository in parent folder of project
选择此复选框可以使用当前工程所在的
workspace目录创建一个新的版本仓库
Project
Location
Repository
testGit F:\architectWorkspace\testGit
Create Repository
F:\architectWorkspace\testGit
\.git
创建成功后，可以点击Finish按钮完成新增内容
Finish
Cancel

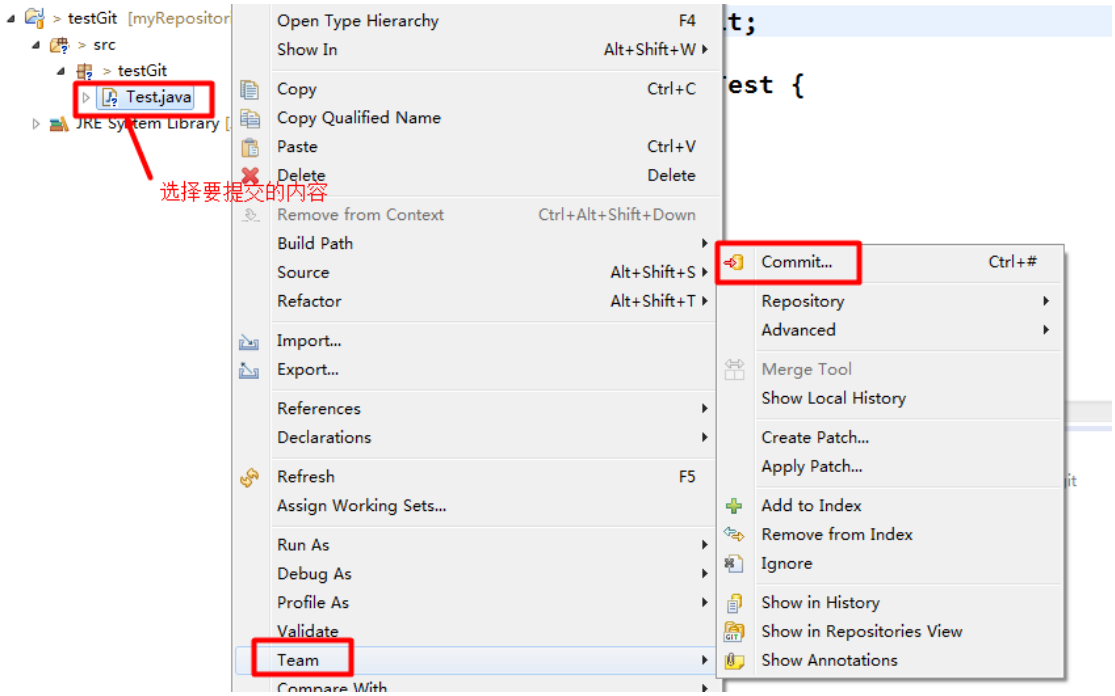

Package Explorer
testGit [myRepositories master]
src
JRE System Library [JavaSE-1.7]
New
Go Into
Open in New Window
Open Type Hierarchy F4
Show In Alt+Shift+W
Copy Ctrl+C
Copy Qualified Name
Paste Ctrl+V
Delete Delete
Remove from Context Ctrl+Alt+Shift+Down
Build Path
Source Alt+Shift+S
Refactor Alt+Shift+T
Import...
Export...
Refresh F5
Close Project
Close Unrelated Projects
Assign Working Sets...
Run As
Debug As
Profile As
Validate
Team
Compare With
提交新增内容
Commit... Ctrl+#
Stashes
Push to Upstream
Fetch from Upstream
Push Branch 'master'...
Pull
Remote
Switch To
Advanced
Synchronize Workspace
Merge Tool
Merge...
Rebase...
Reset...
Create Patch...
Apply Patch...
Add to Index
Remove from Index
Ignore
Show in History
Show in Repositories View
Disconnect

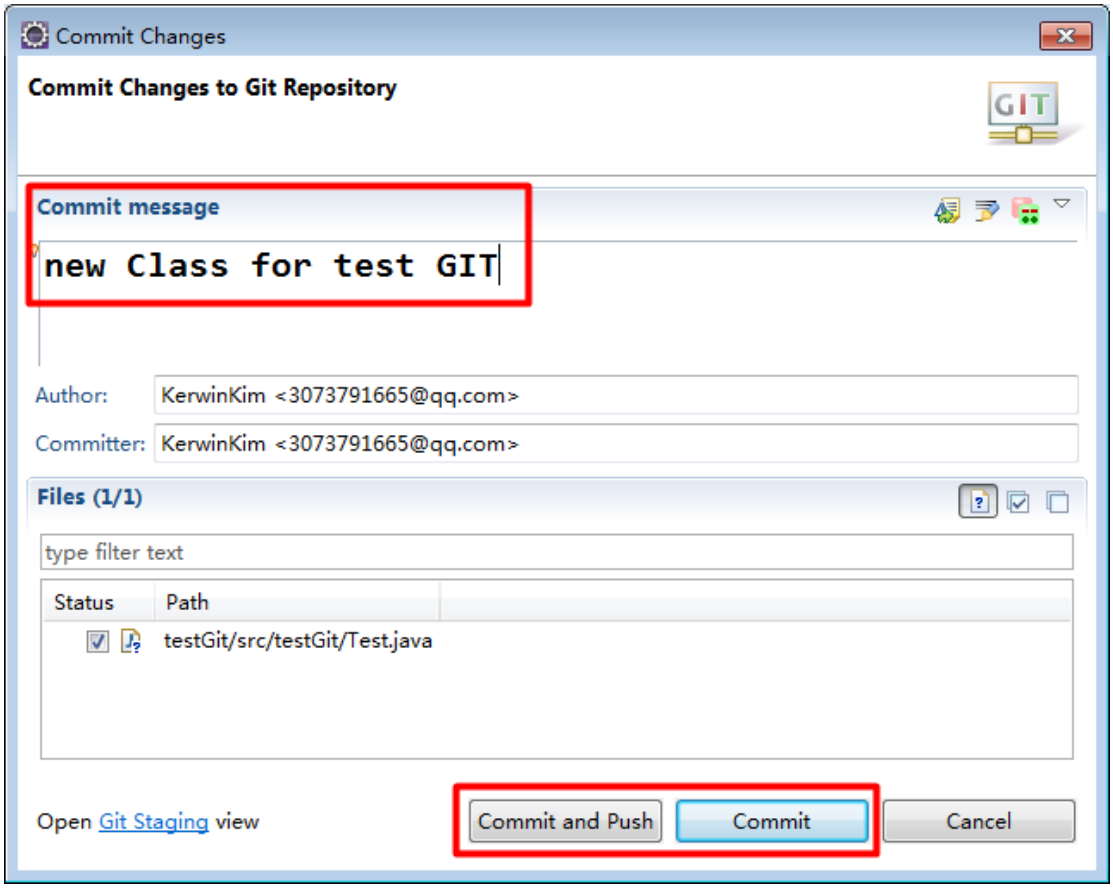

Commit Changes
Commit Changes to Git Repository
GIT
Commit message
new project for test git
提交操作注释
Author: KerwinKim <3073791665@qq.com>
Committer: KerwinKim <3073791665@qq.com>
Files (4/4)
选择要提交的内容
type filter text
Status
Path
testGit/.classpath
testGit/.gitignore
testGit/.project
testGit/.settings/org.eclipse.jdt.core.prefs
仅提交到本地版本仓库
提交到本地版本仓库并同时推送到远程版本仓库
Open Git Staging view
Commit and Push
Commit
Cancel





7 提交内容

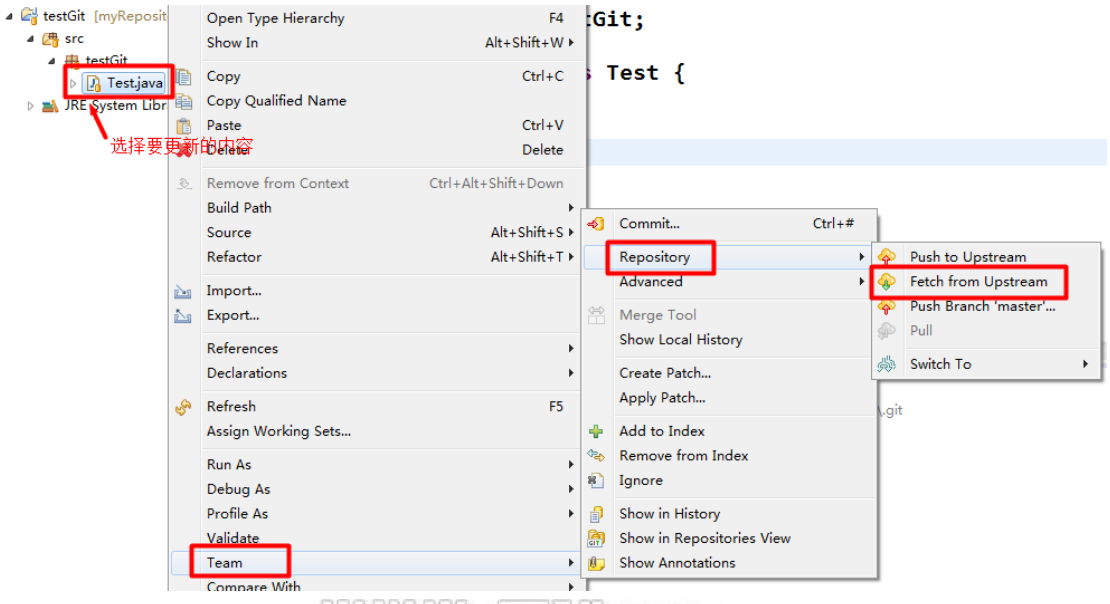




8 更新内容

8.1 fetch

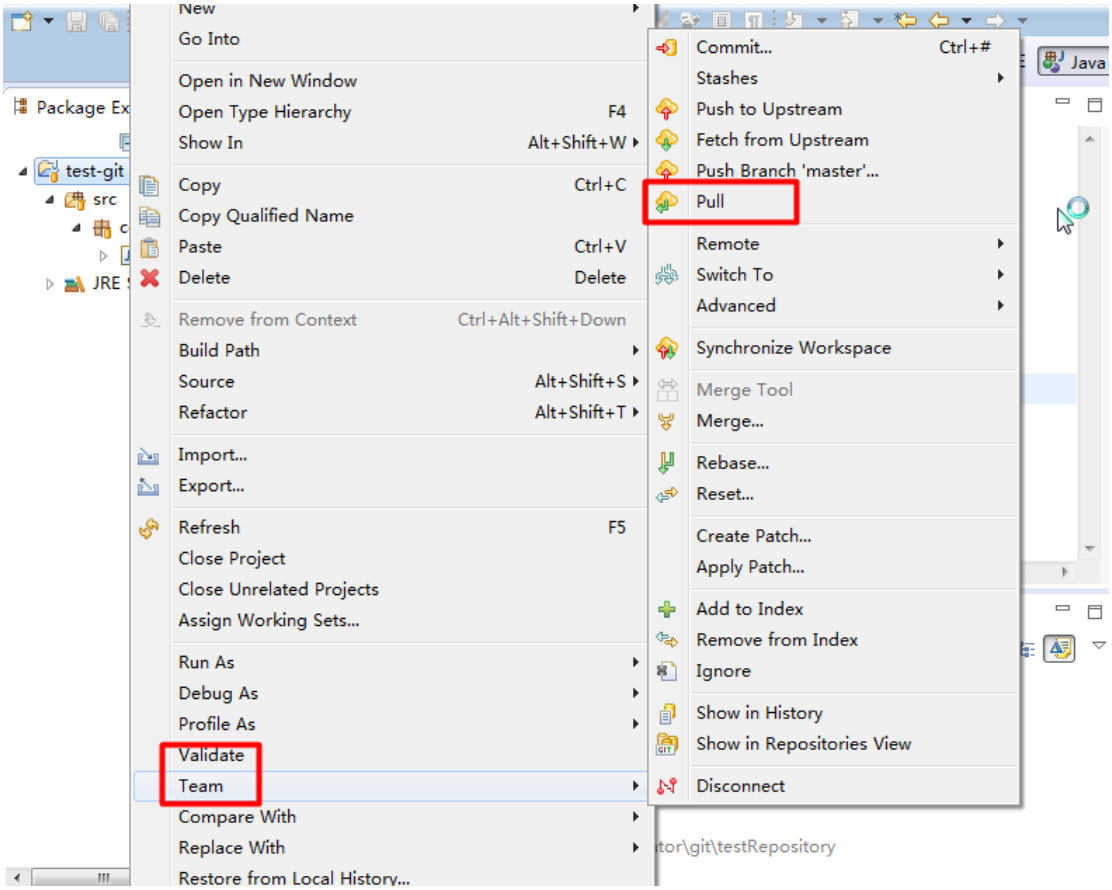
抓取， 抓取远程的 *head* 信息。从远程仓库中下载 *head* 头信息的变更状态。没有下载真实的代码变化。



8.2 pull

拉取，相当于先 *fetch head* 再 *pull code*。 先下载 *head* 头的变更，再根据头信息的变更下载真实的代码。

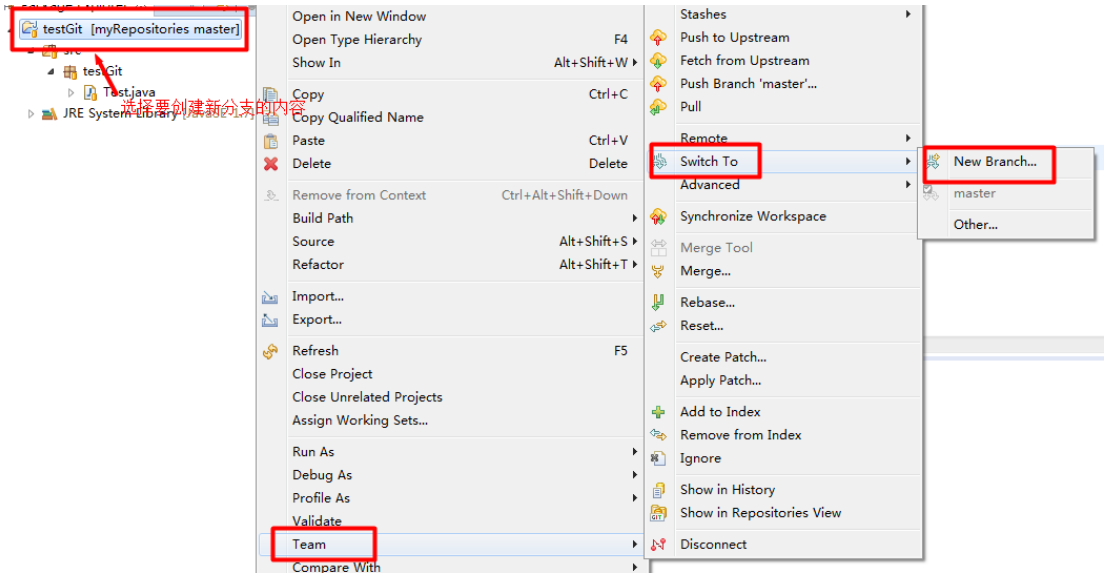
通常直接 *pull* 即可。

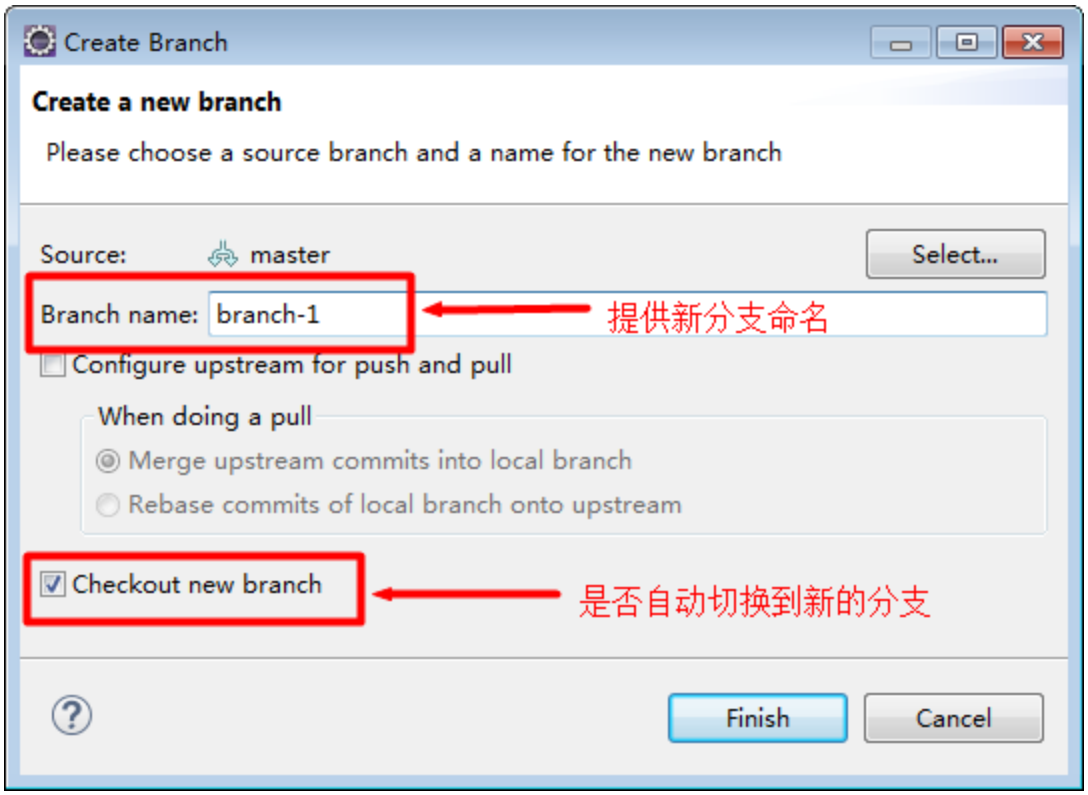


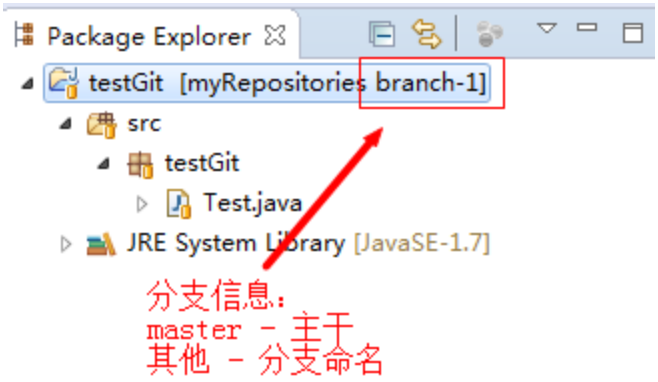
9 分支管理

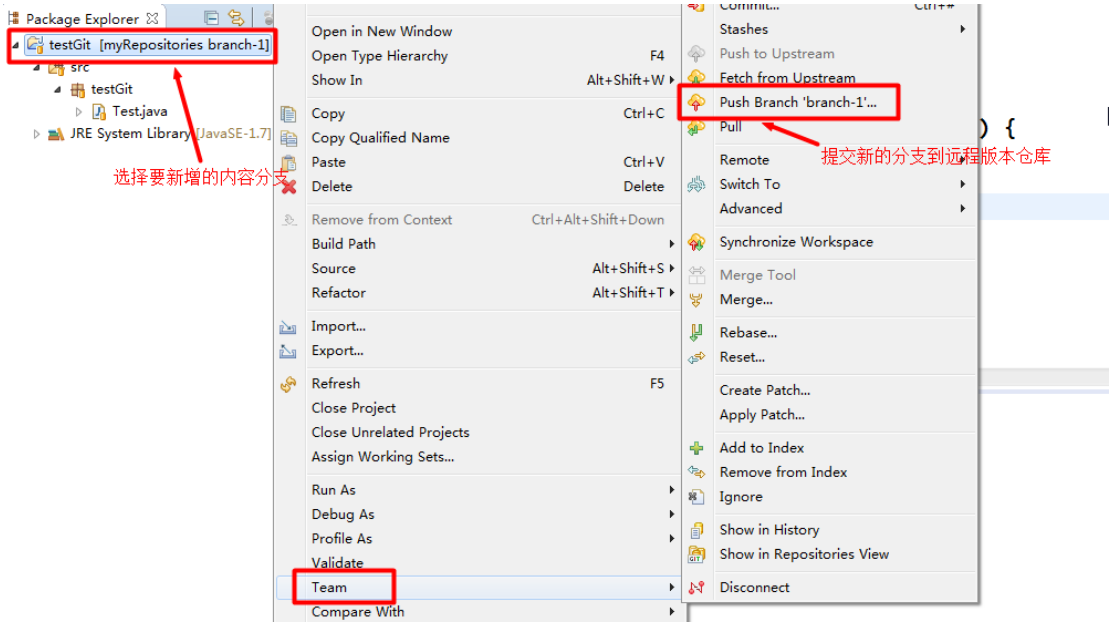
分支管理，是 *GIT* 开发中的一个非常有效的团队开发策略。多个程序员并行开发，每个程序员可以定义各自的分支，在自己的分支上开发工程。再开发结束测试完毕后，再合并到主干工程中，一次性提交到远程。由其他程序员使用。

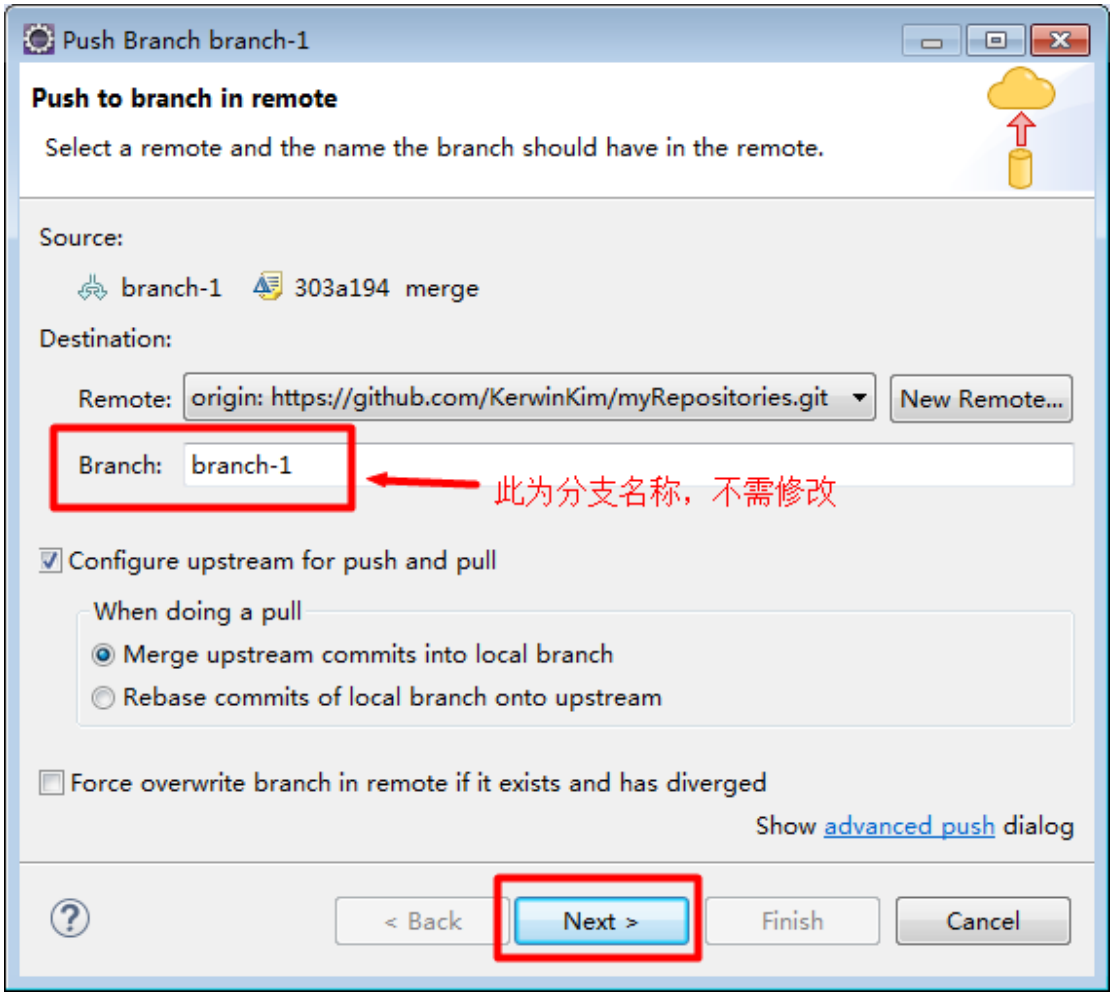
9.1 创建新分支

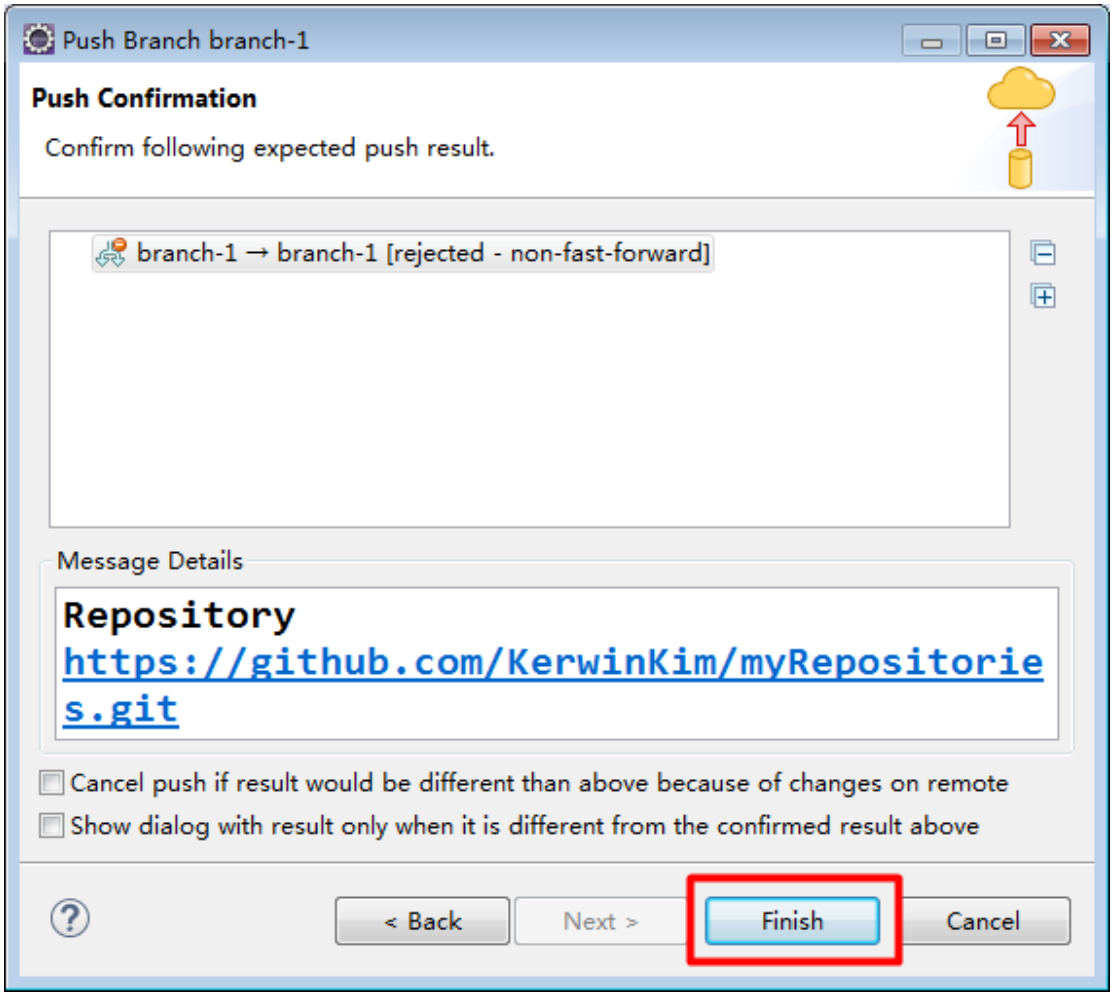

testGit [myRepositories master]
选择要创建新分支的内容
Open in New Window
Open Type Hierarchy F4
Show In Alt+Shift+W
Copy Ctrl+C
Copy Qualified Name
Paste Ctrl+V
Delete Delete
Remove from Context Ctrl+Alt+Shift+Down
Build Path
Source Alt+Shift+S
Refactor Alt+Shift+T
Import...
Export...
Refresh F5
Close Project
Close Unrelated Projects
Assign Working Sets...
Run As
Debug As
Profile As
Validate
Team
Compare With
Stashes
Push to Upstream
Fetch from Upstream
Push Branch 'master'...
Pull
Remote
Switch To
Advanced
Synchronize Workspace
Merge Tool
Merge...
Rebase...
Reset...
Create Patch...
Apply Patch...
Add to Index
Remove from Index
Ignore
Show in History
Show in Repositories View
Disconnect
New Branch...
master
Other...


Create Branch
Create a new branch
Please choose a source branch and a name for the new branch
Source: master
Select...
Branch name: branch-1
提供新分支命名
Configure upstream for push and pull
When doing a pull
Merge upstream commits into local branch
Rebase commits of local branch onto upstream
Checkout new branch
是否自动切换到新的分支
Finish
Cancel


Package Explorer
testGit [myRepositories branch-1]
src
testGit
Test.java
JRE System Library [JavaSE-1.7]
分支信息:
master - 主干
其他 - 分支命名

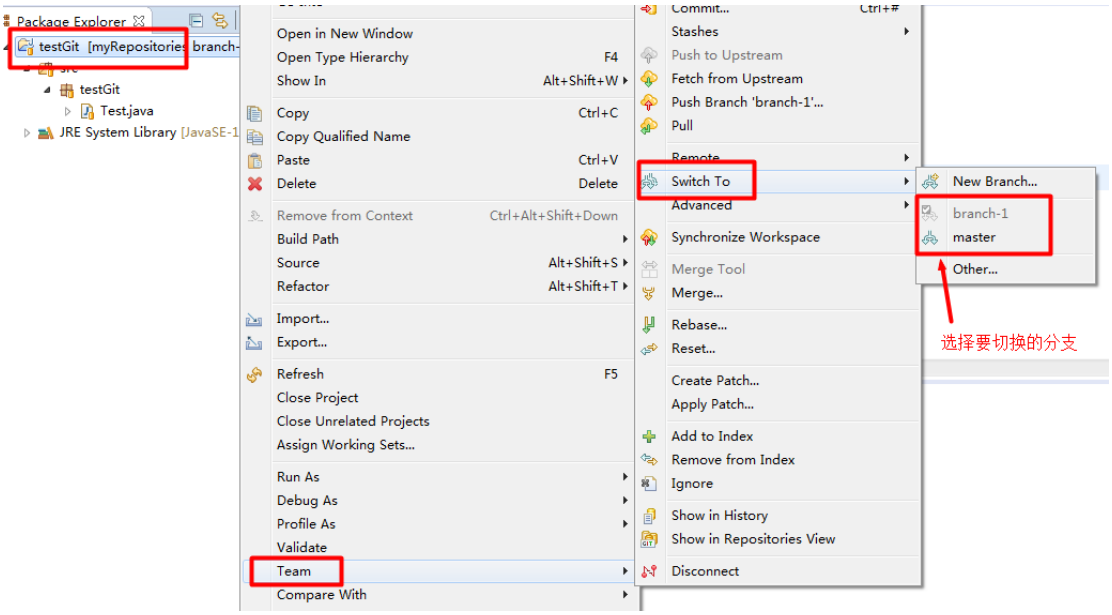

Package Explorer
testGit [myRepositories branch-1]
src
testGit
Test.java
JRE System Library [JavaSE-1.7]
选择要新增的内容分支
Open in New Window
Open Type Hierarchy F4
Show In Alt+Shift+W
Copy Ctrl+C
Copy Qualified Name
Paste Ctrl+V
Delete Delete
Remove from Context Ctrl+Alt+Shift+Down
Build Path
Source Alt+Shift+S
Refactor Alt+Shift+T
Import...
Export...
Refresh F5
Close Project
Close Unrelated Projects
Assign Working Sets...
Run As
Debug As
Profile As
Validate
Team
Compare With
Stashes
Push to Upstream
Fetch from Upstream
Push Branch 'branch-1'...
Pull
Remote
Switch To
Advanced
Synchronize Workspace
Merge Tool
Merge...
Rebase...
Reset...
Create Patch...
Apply Patch...
Add to Index
Remove from Index
Ignore
Show in History
Show in Repositories View
Disconnect
提交新的分支到远程版本仓库


Push Branch branch-1
Push to branch in remote
Select a remote and the name the branch should have in the remote.
Source:
branch-1
303a194 merge
Destination:
Remote:
origin: https://github.com/KerwinKim/myRepositories.git
New Remote...
Branch:
branch-1
此为分支名称，不需修改
Configure upstream for push and pull
When doing a pull
Merge upstream commits into local branch
Rebase commits of local branch onto upstream
Force overwrite branch in remote if it exists and has diverged
Show advanced push dialog
< Back
Next >
Finish
Cancel


Push Branch branch-1
Push Confirmation
Confirm following expected push result.
branch-1 → branch-1 [rejected - non-fast-forward]
Message Details
Repository
https://github.com/KerwinKim/myRepositories.git
Cancel push if result would be different than above because of changes on remote
Show dialog with result only when it is different from the confirmed result above
< Back
Next >
Finish
Cancel

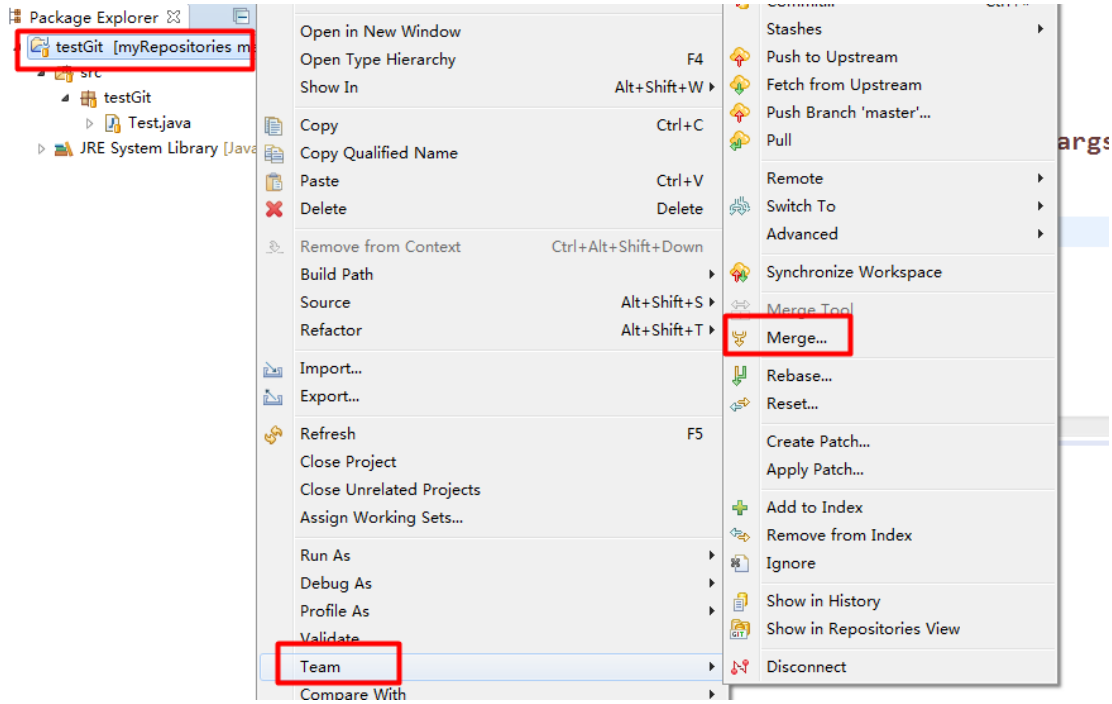

myRepositories [branch-1 ↑1 ↓1] - C:\Users\Administrator\git\myRepositories\.git
Branches
Local
branch-1 303a194 merge
master 303a194 merge
Remote Tracking
origin/branch-1 05e4d62 branch commit test
origin/master 303a194 merge

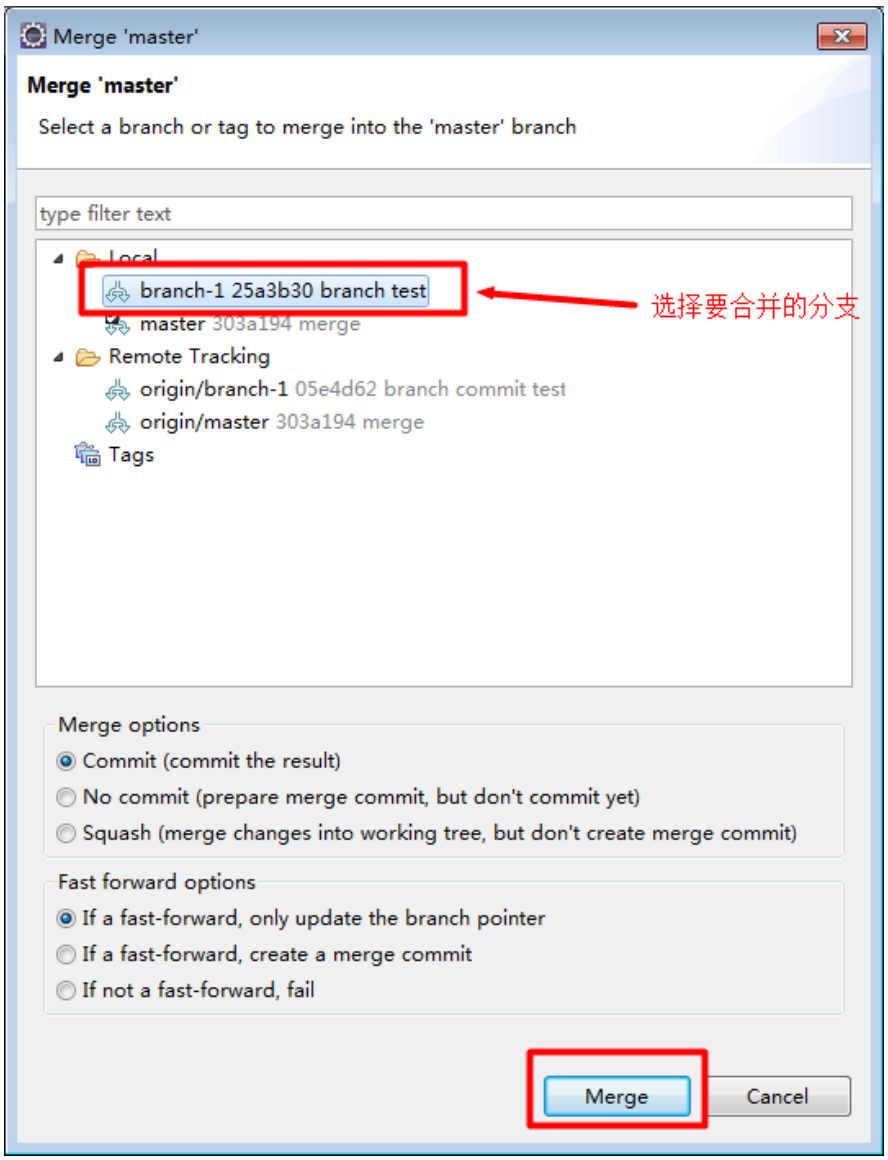
9.2 分支切换

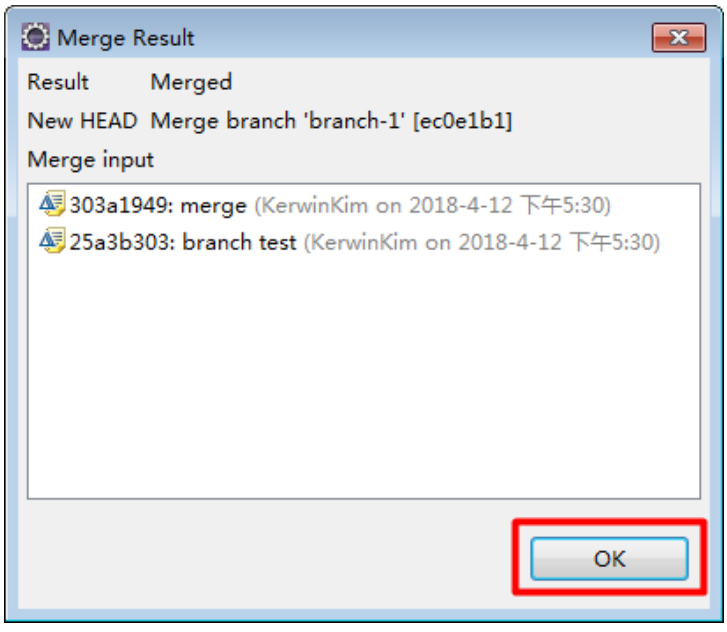


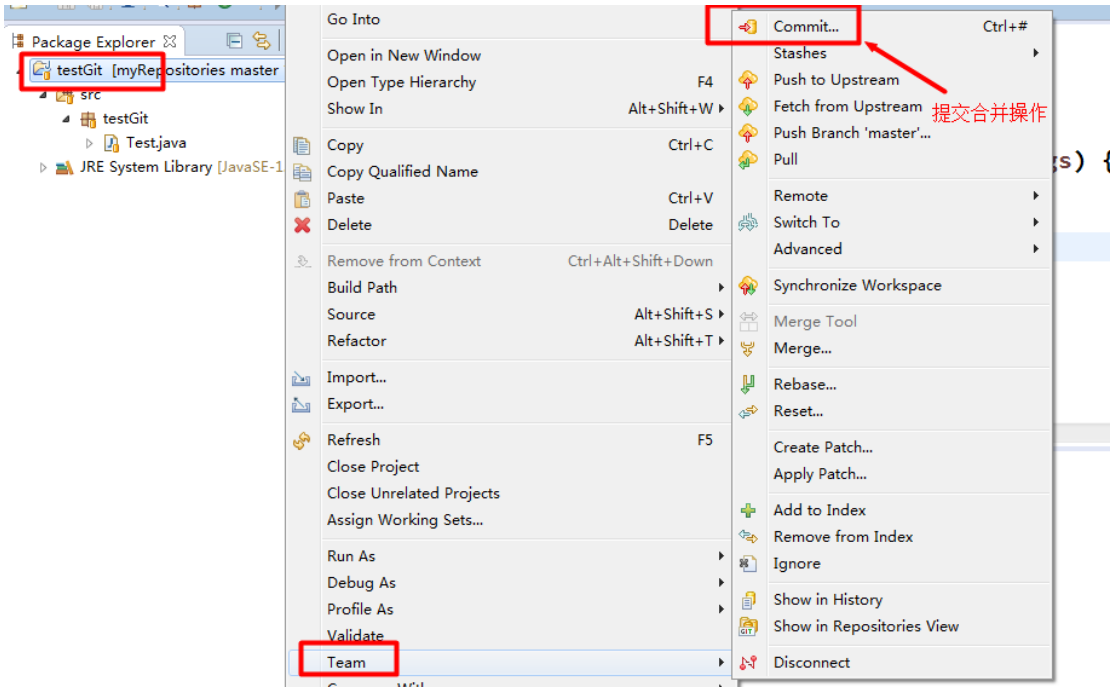
9.3 合并分支

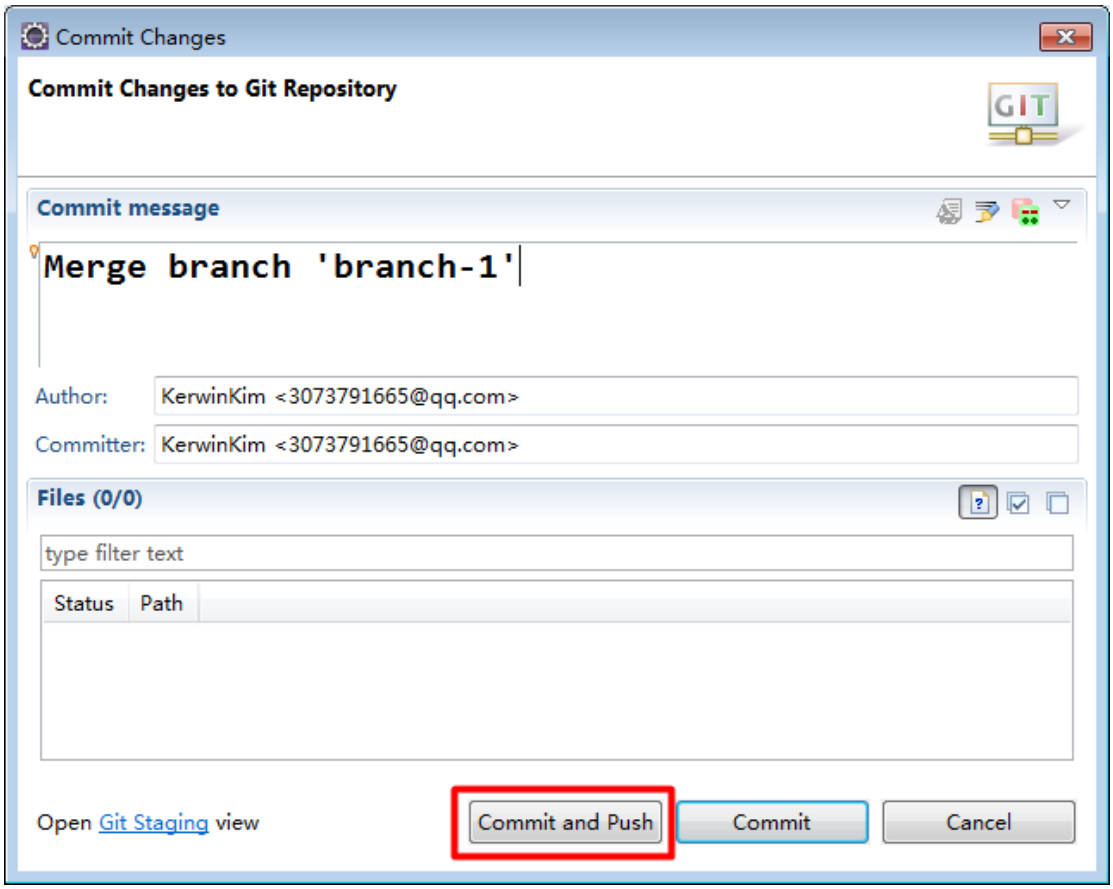
首先切换到 *master* 主干，再执行下述操作：




Merge 'master'
Merge 'master'
Select a branch or tag to merge into the 'master' branch
type filter text
Local
branch-1 25a3b30 branch test
master 303a194 merge
Remote Tracking
origin/branch-1 05e4d62 branch commit test
origin/master 303a194 merge
Tags
选择要合并的分支
Merge options
Commit (commit the result)
No commit (prepare merge commit, but don't commit yet)
Squash (merge changes into working tree, but don't create merge commit)
Fast forward options
If a fast-forward, only update the branch pointer
If a fast-forward, create a merge commit
If not a fast-forward, fail
Merge
Cancel

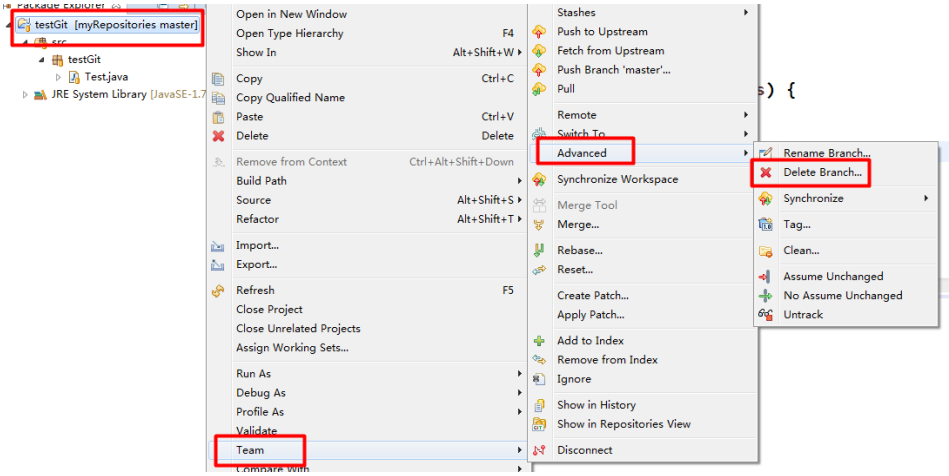

Merge Result
Result Merged
New HEAD Merge branch 'branch-1' [ec0e1b1]
Merge input
303a1949: merge (KerwinKim on 2018-4-12 下午5:30)
25a3b303: branch test (KerwinKim on 2018-4-12 下午5:30)
OK

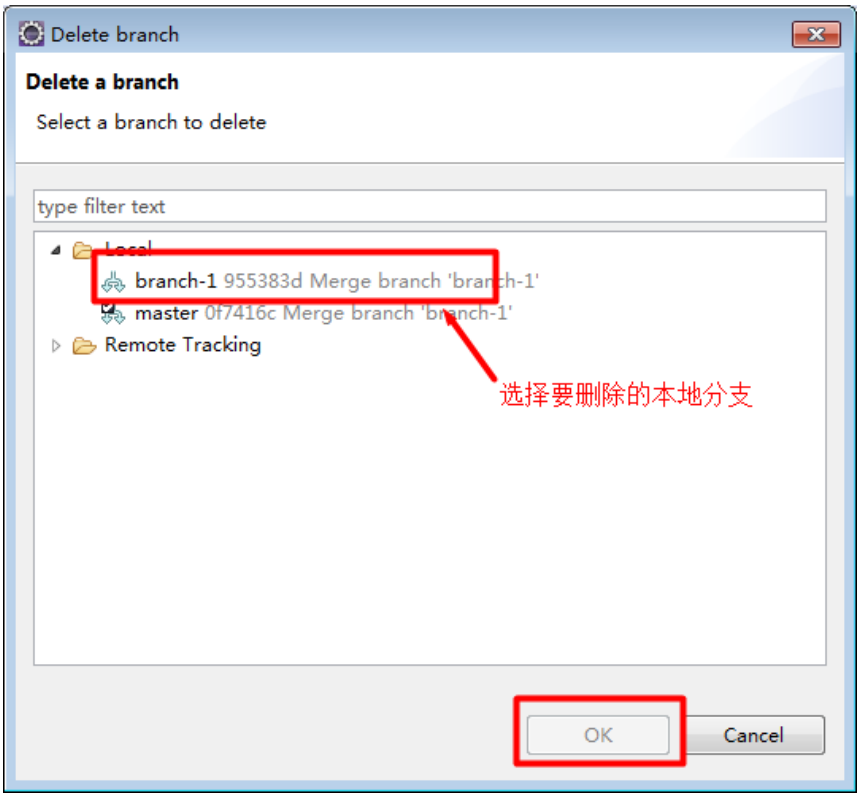

Package Explorer
testGit [myRepositories master
src
testGit
Test.java
JRE System Library [JavaSE-1
Go Into
Open in New Window
Open Type Hierarchy F4
Show In Alt+Shift+W
Copy Ctrl+C
Copy Qualified Name
Paste Ctrl+V
Delete Delete
Remove from Context Ctrl+Alt+Shift+Down
Build Path
Source Alt+Shift+S
Refactor Alt+Shift+T
Import...
Export...
Refresh F5
Close Project
Close Unrelated Projects
Assign Working Sets...
Run As
Debug As
Profile As
Validate
Team
Commit... Ctrl+#
Stashes
Push to Upstream
Fetch from Upstream
Push Branch 'master'...
Pull
Remote
Switch To
Advanced
Synchronize Workspace
Merge Tool
Merge...
Rebase...
Reset...
Create Patch...
Apply Patch...
Add to Index
Remove from Index
Ignore
Show in History
Show in Repositories View
Disconnect
提交合并操作

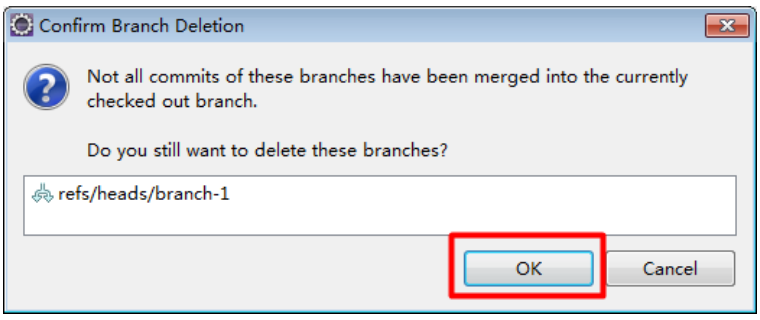

Commit Changes
Commit Changes to Git Repository
GIT
Commit message
Merge branch 'branch-1'
Author: KerwinKim <3073791665@qq.com>
Committer: KerwinKim <3073791665@qq.com>
Files (0/0)
type filter text
Status
Path
Open Git Staging view
Commit and Push
Commit
Cancel

9.4 删除分支

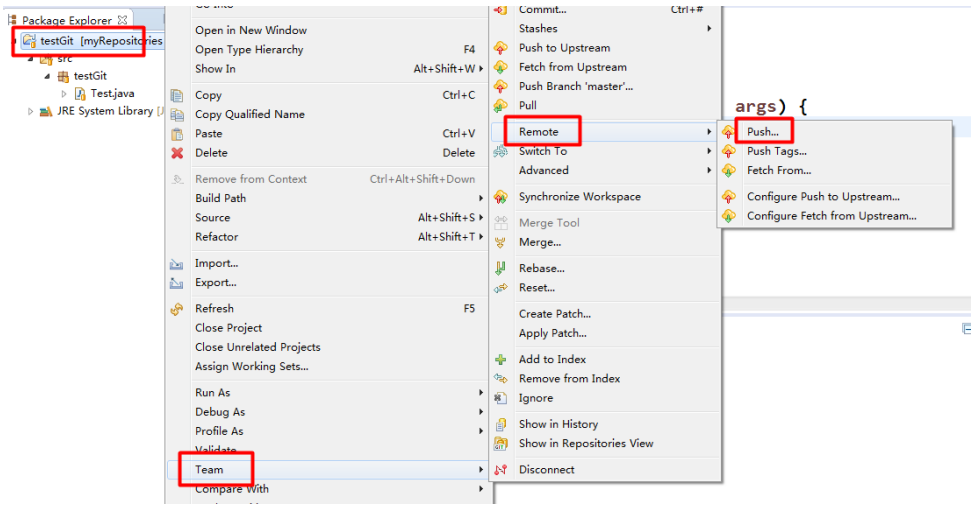
9.4.1 删除本地分支

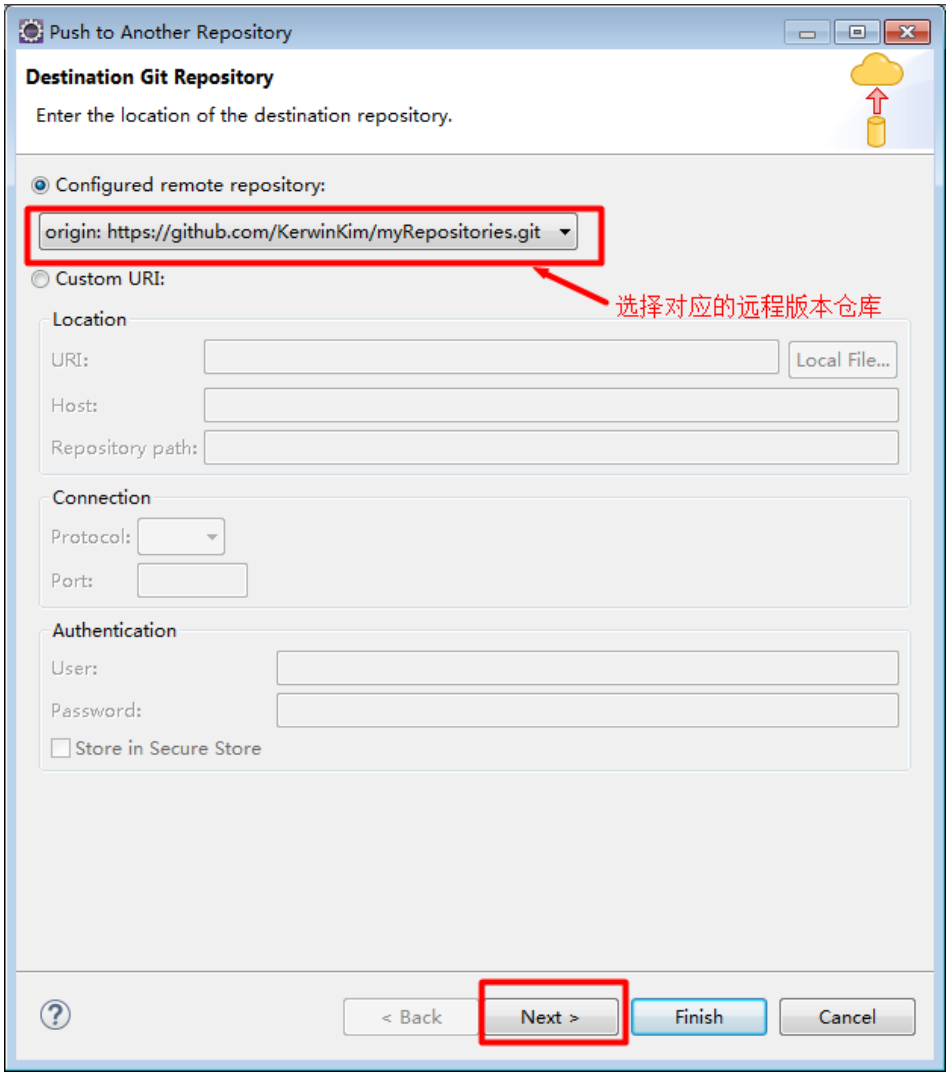


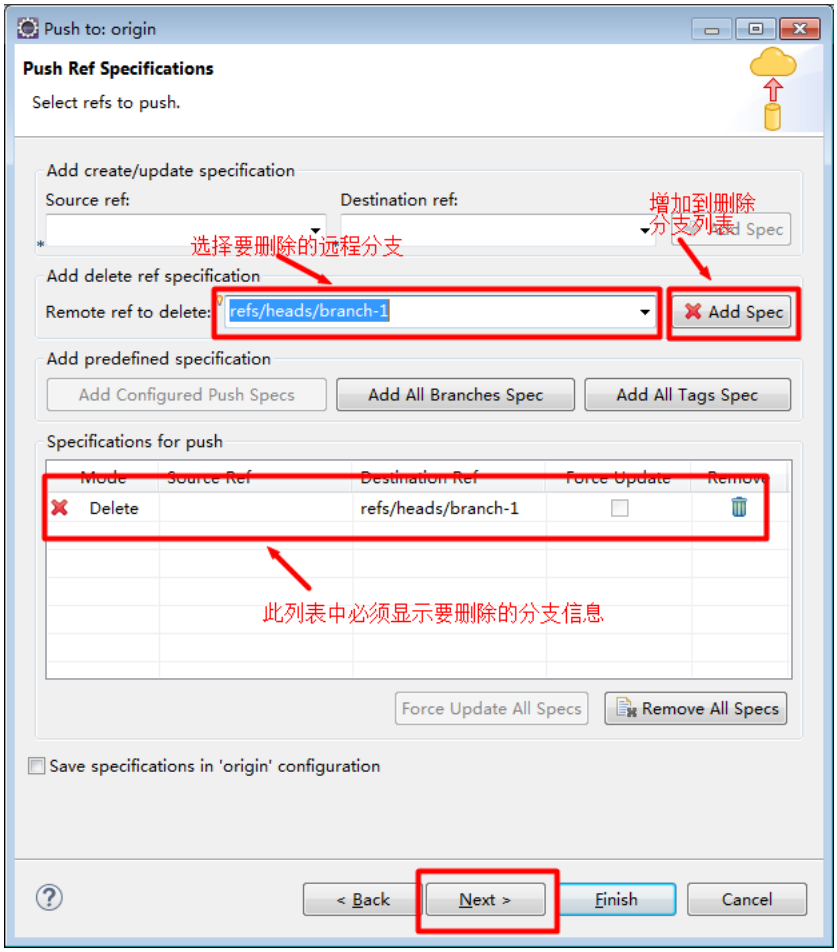


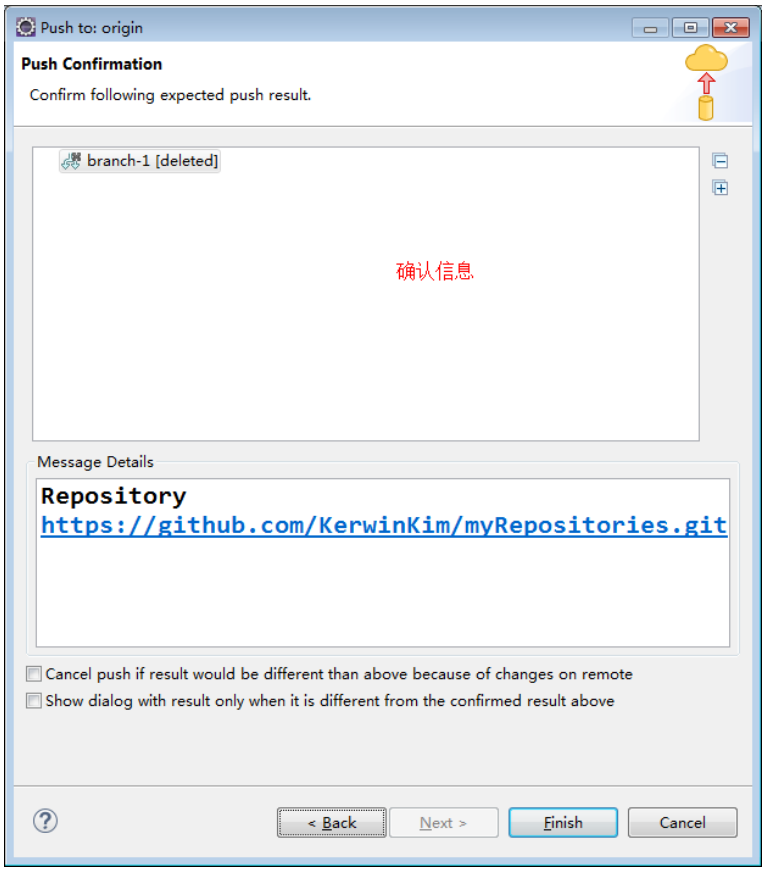


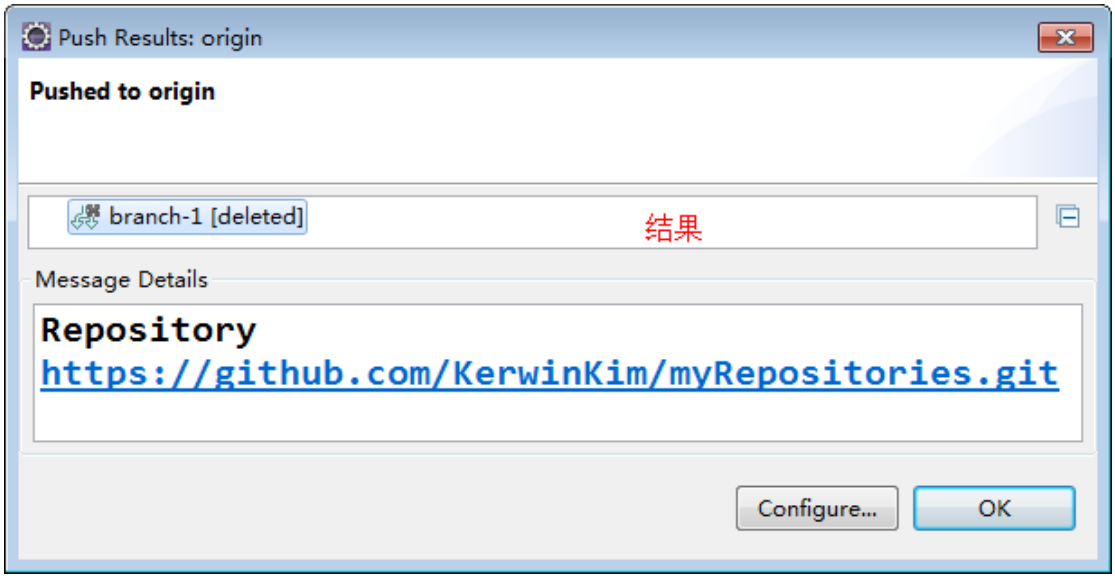
9.4.2 删除远程分支






Push to: origin
Push Ref Specifications
Select refs to push.
Add create/update specification
Source ref:
Destination ref:
增加到删除分支列表
选择要删除的远程分支
Add delete ref specification
Remote ref to delete:
refs/heads/branch-1
Add Spec
Add predefined specification
Add Configured Push Specs
Add All Branches Spec
Add All Tags Spec
Specifications for push
Mode
Source Ref
Destination Ref
Force Update
Remove
Delete
refs/heads/branch-1
此列表中必须显示要删除的分支信息
Force Update All Specs
Remove All Specs
Save specifications in 'origin' configuration
< Back
Next >
Finish
Cancel


Push to: origin
Push Confirmation
Confirm following expected push result.
branch-1 [deleted]
确认信息
Message Details
Repository
https://github.com/KerwinKim/myRepositories.git
Cancel push if result would be different than above because of changes on remote
Show dialog with result only when it is different from the confirmed result above
< Back
Next >
Finish
Cancel



10 冲突管理

10.1提交冲突

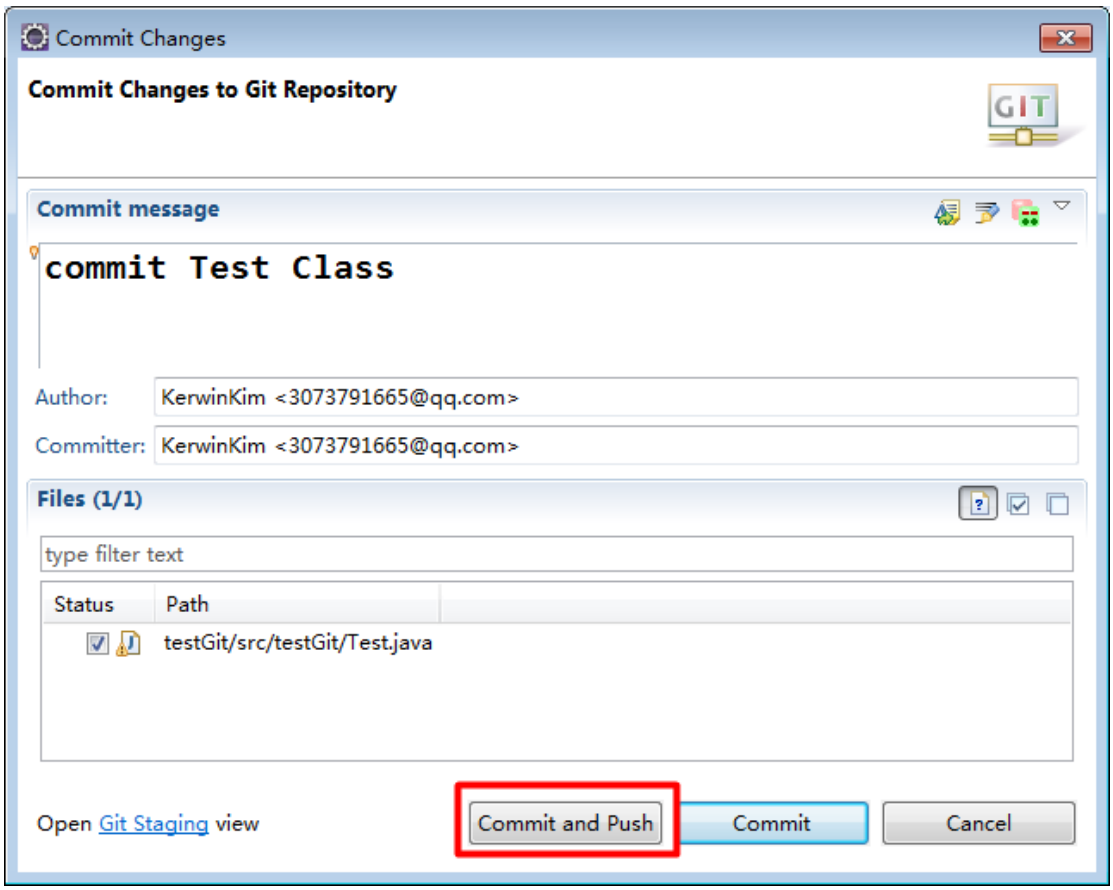
10.1.1 远程代码

```
1 package testGit;
2
3 public class Test {
4     public static void main(String[] args) {
5         System.out.println();
6     }
7
8 }
```

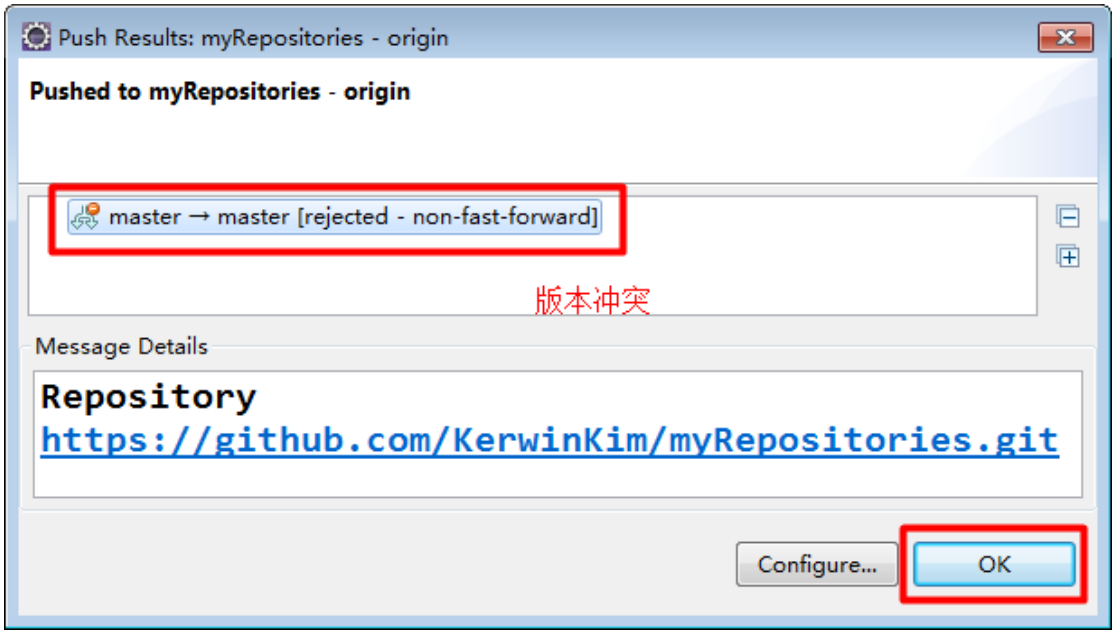
10.1.2 本地代码

```
1 package testGit;
2
3 public class Test {
4     public static void main(String[] args) {
5         int i = 0;
6     }
7
8 }
9
```

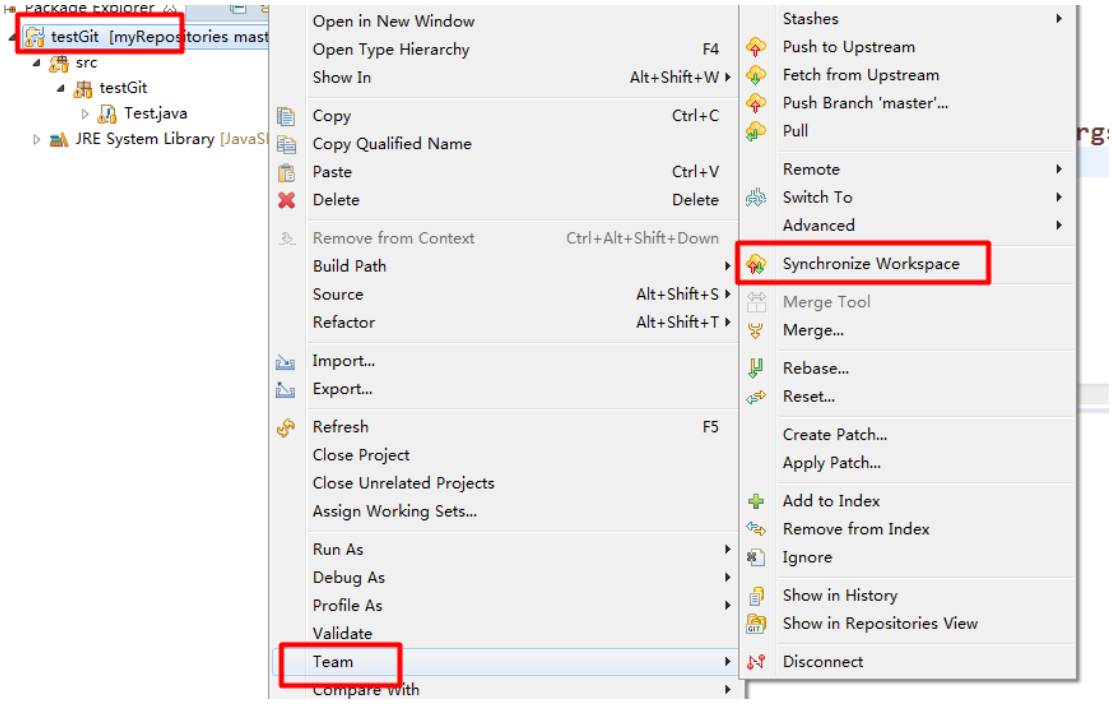
10.1.3 提交

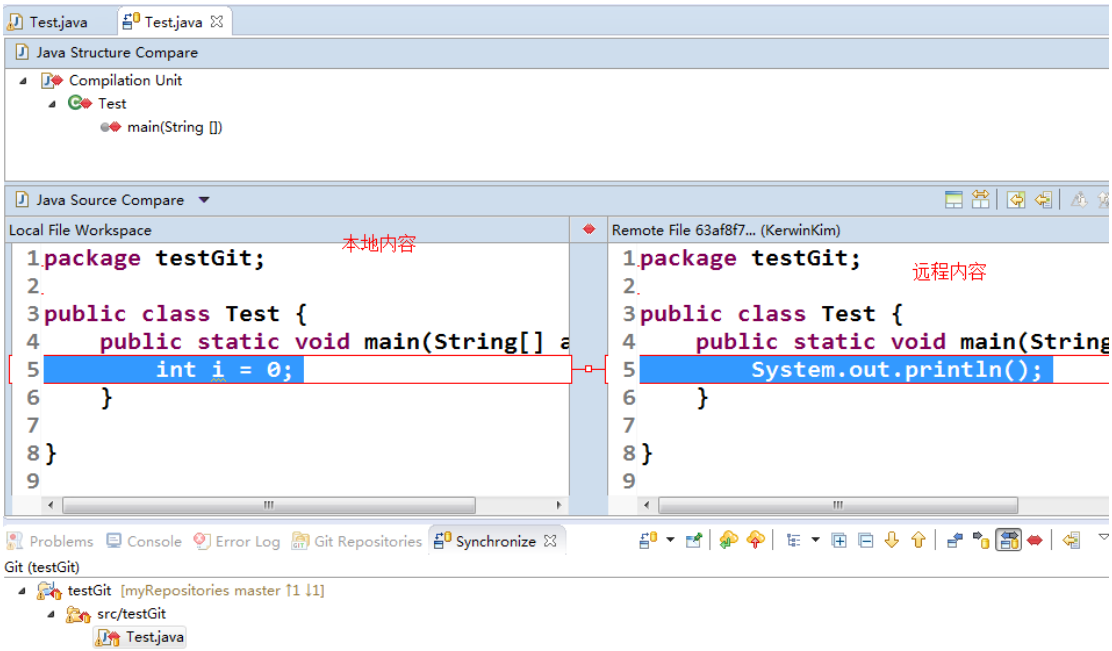


10.1.4 结果

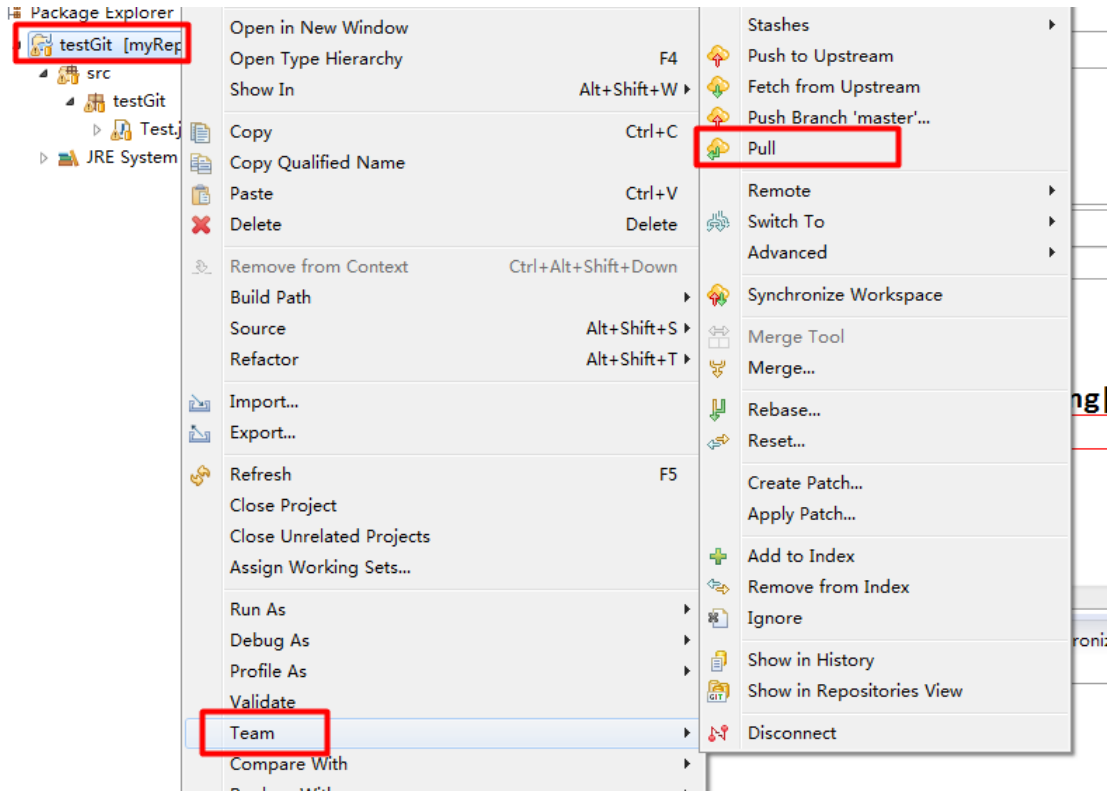


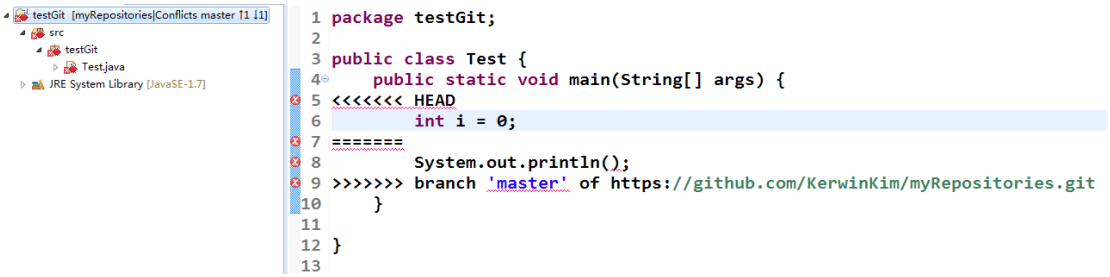
10.2同步



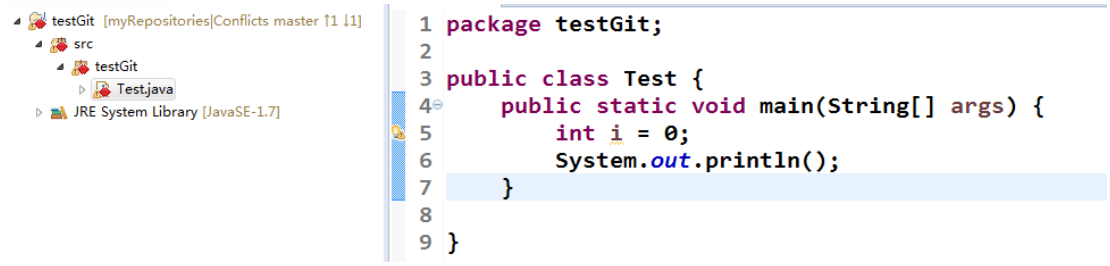


10.3pull 远程代码

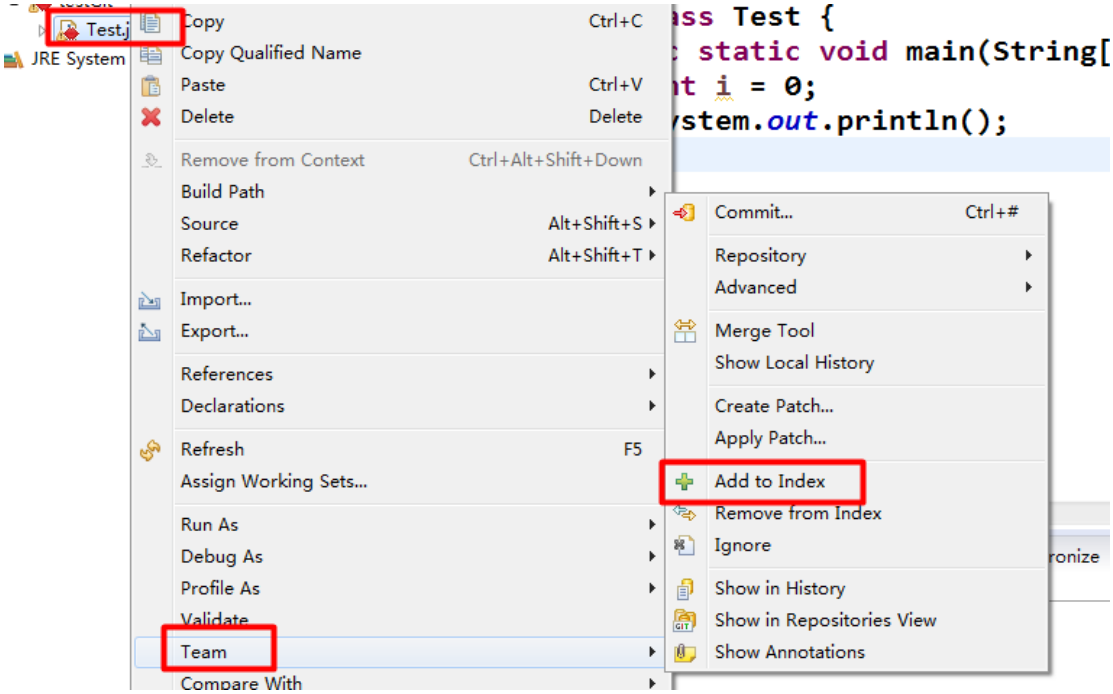


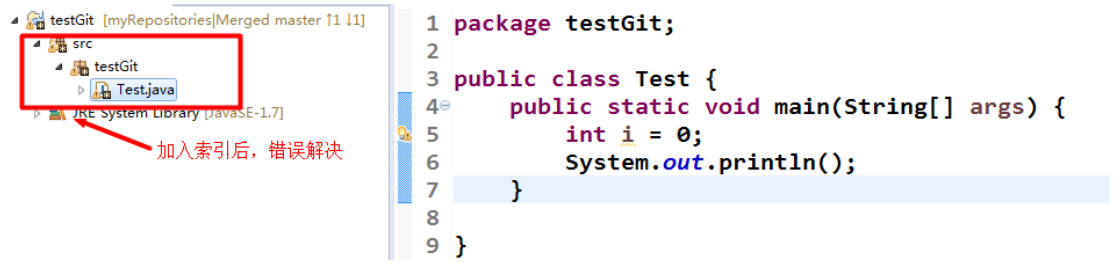


10.4修改内容

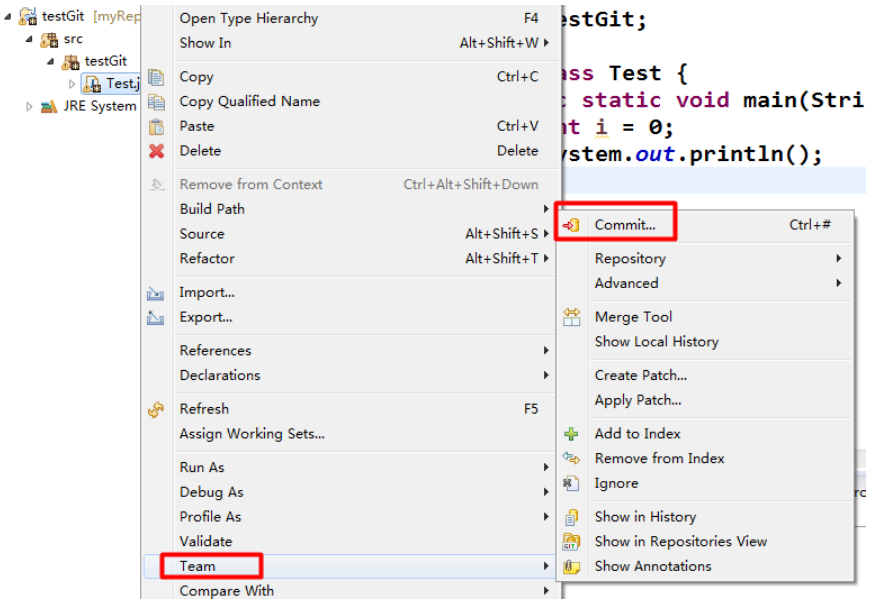


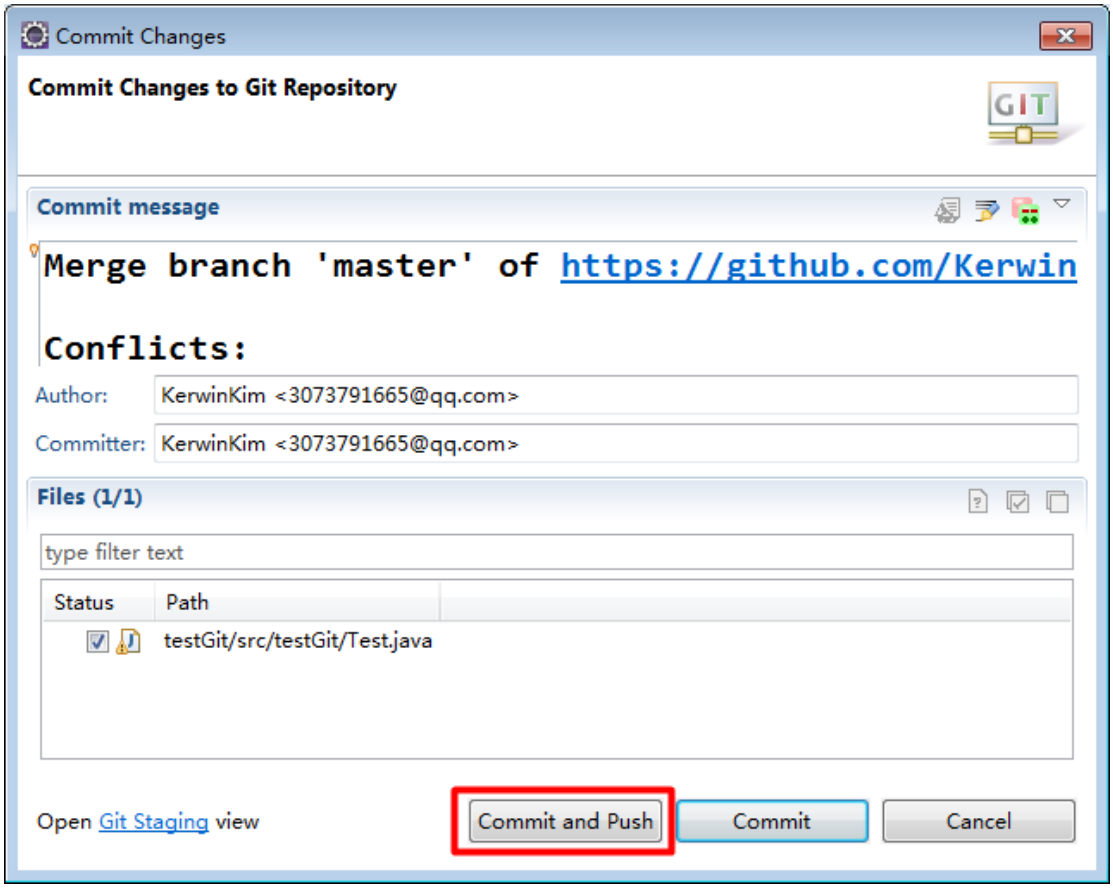
10.5将修改后的内容加入索引信息

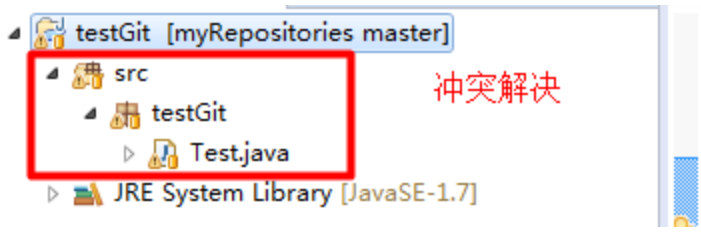




10.6提交内容

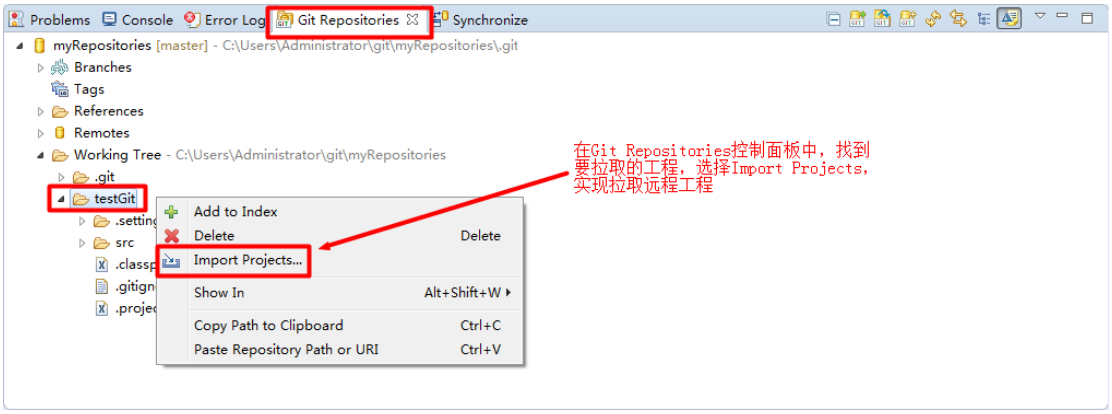


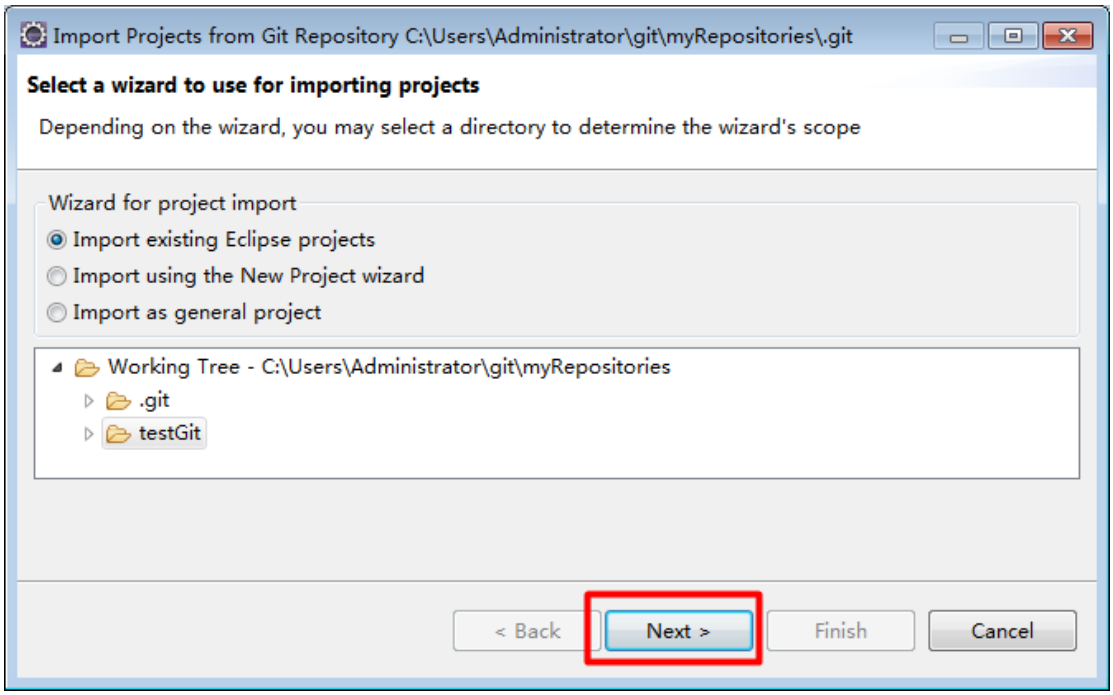


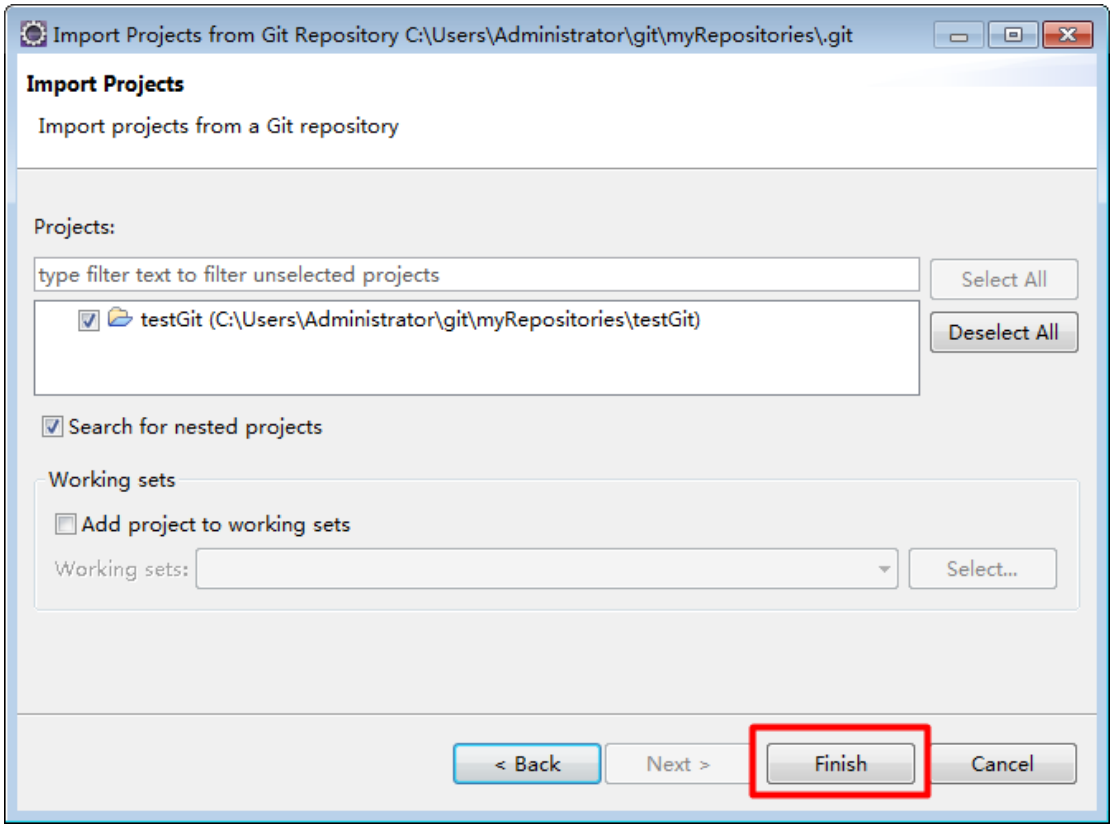


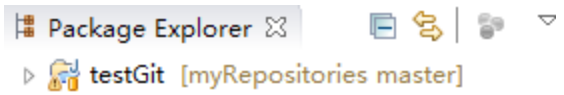
11 拉取内容

11.1拉取远程工程

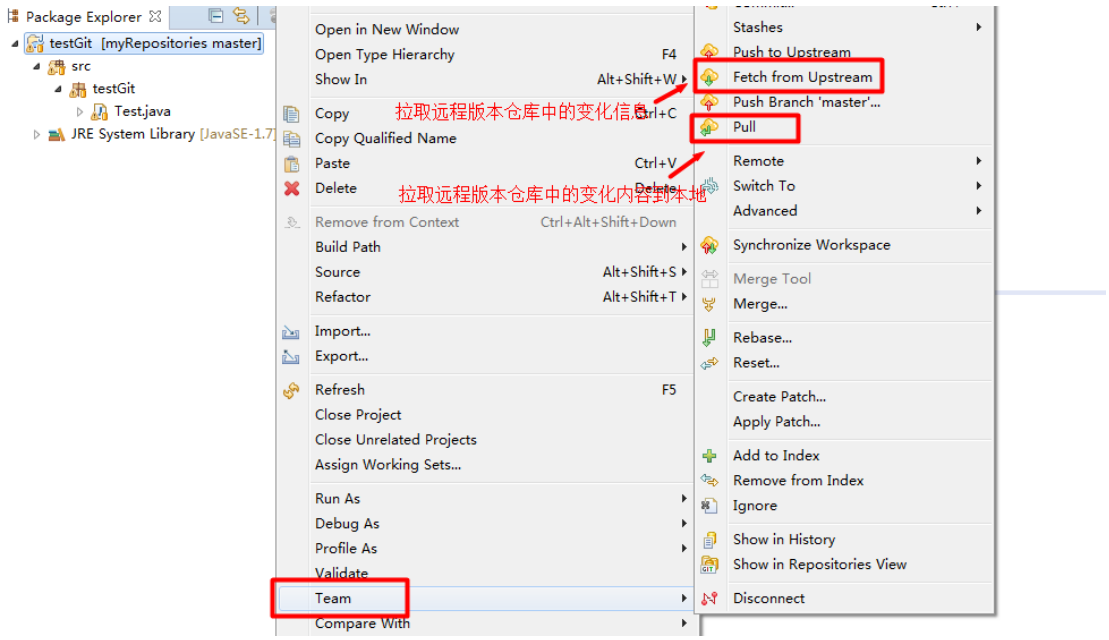



Import Projects from Git Repository C:\Users\Administrator\git\myRepositories\.git
Select a wizard to use for importing projects
Depending on the wizard, you may select a directory to determine the wizard's scope
Wizard for project import
Import existing Eclipse projects
Import using the New Project wizard
Import as general project
Working Tree - C:\Users\Administrator\git\myRepositories
.git
testGit
< Back
Next >
Finish
Cancel

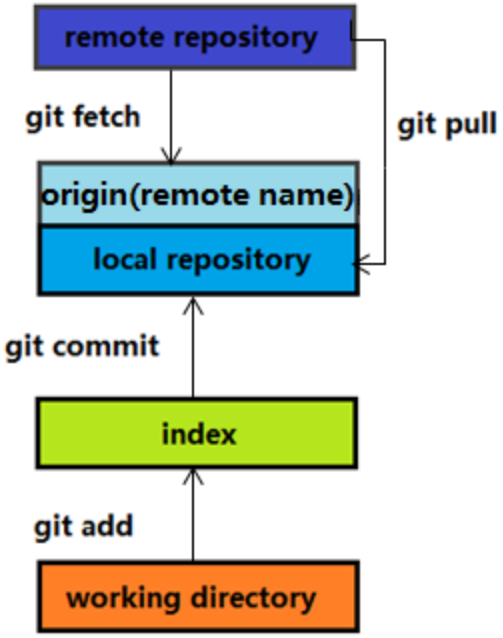

Import Projects from Git Repository C:\Users\Administrator\git\myRepositories\.git
Import Projects
Import projects from a Git repository
Projects:
type filter text to filter unselected projects
Select All
testGit (C:\Users\Administrator\git\myRepositories\testGit)
Deselect All
Search for nested projects
Working sets
Add project to working sets
Working sets:
Select...
< Back
Next >
Finish
Cancel


Package Explorer
testGit [myRepositories master]

11.2拉取远程代码

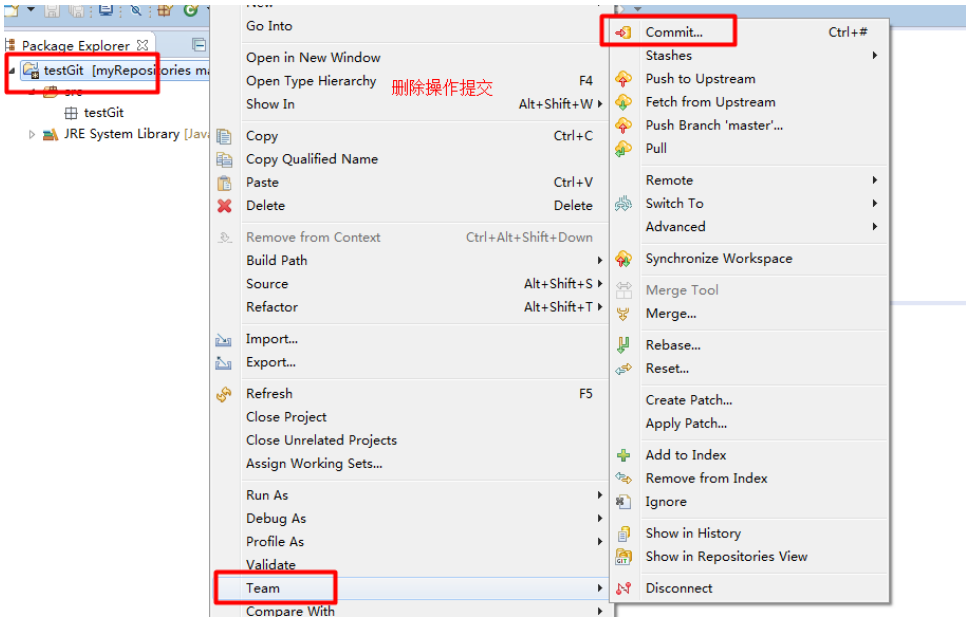


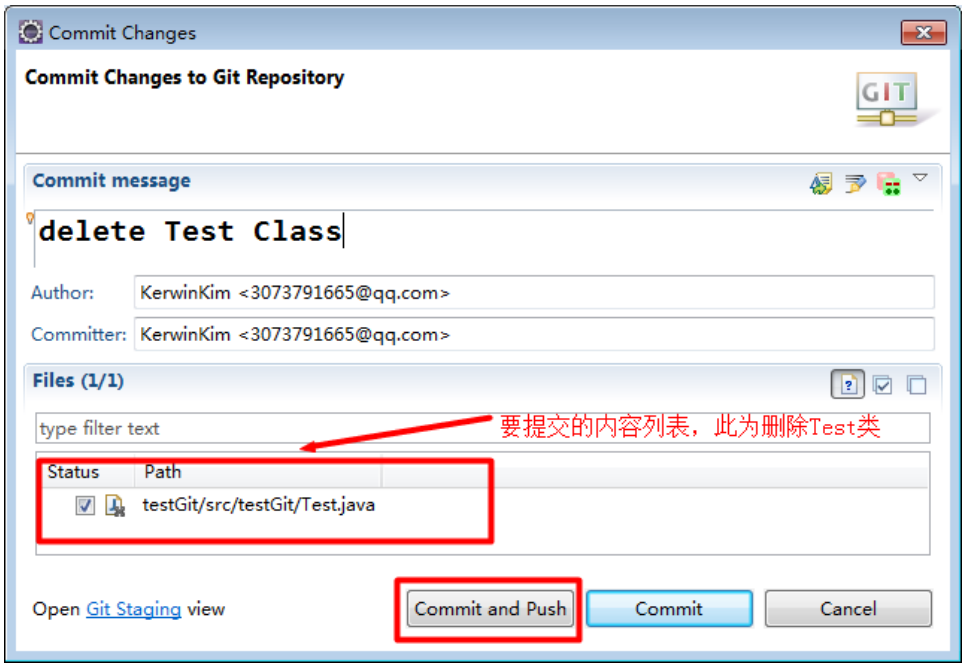
在 *Git* 中，*fetch* 和 *pull* 的功能不同，*fetch* 是拉取远程版本库中的变化信息，是将远程版本仓库中最新版本的 *heads* 头信息更新到本地版本仓库，不包含具体的内容。*pull* 则是将远程版本仓库中的 *heads* 和内容都更新到本地仓库。可用下述图例简单描述：



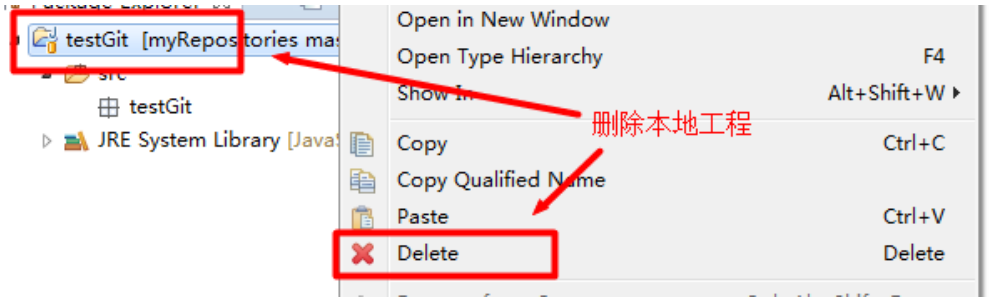
12 删除内容

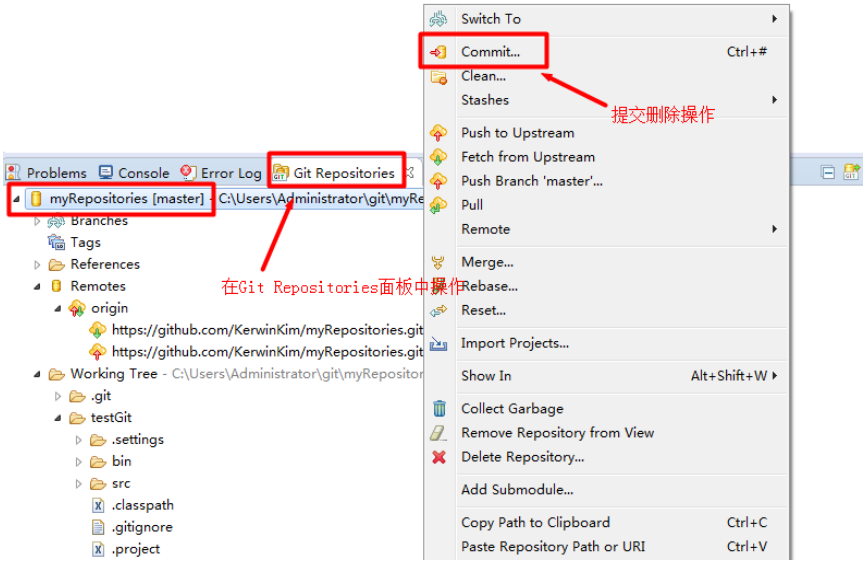
12.1删除代码

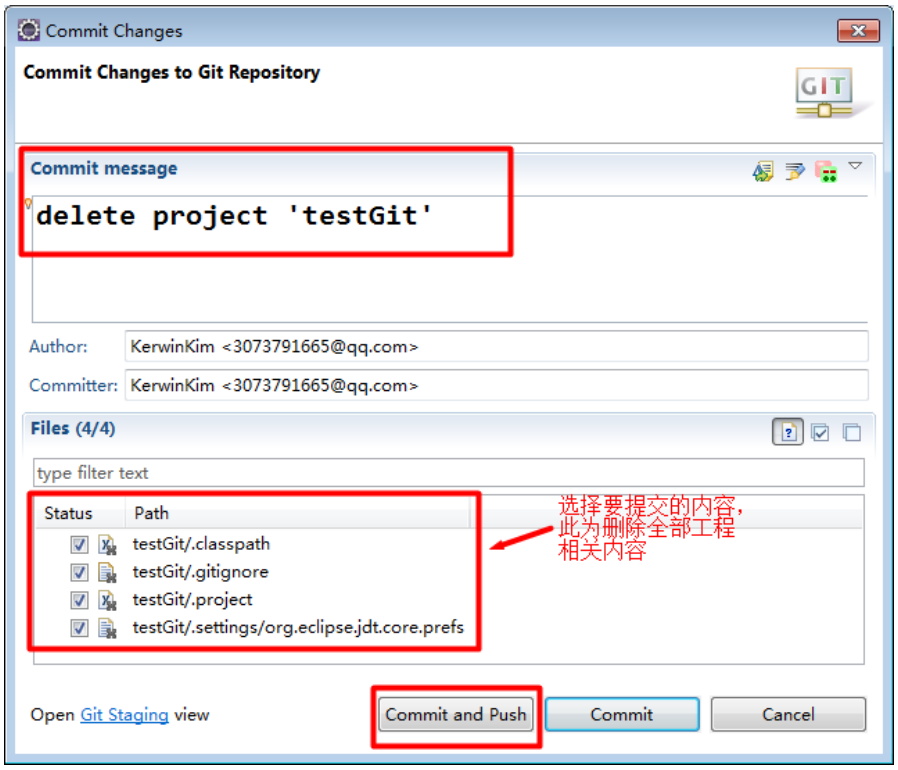




12.2删除工程







13 忽略文件

如果本地有若干文件或目录不想提交到远程。可以使用忽略的方式实现。

.gitignore 文件中定义的内容，都是忽略的内容。不上传到远程仓库的内容。

忽略信息定义方式：

目录 -/目录名称/

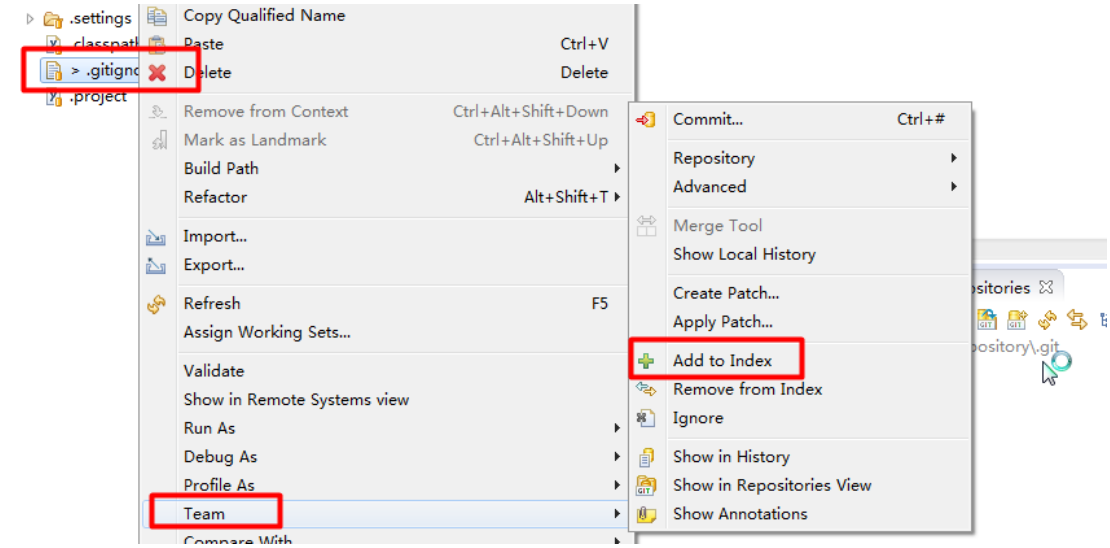
文件 -/文件名称 、 /目录名称/文件名称

匹配 -/前缀* 、 /*后缀 、 /*中缀*

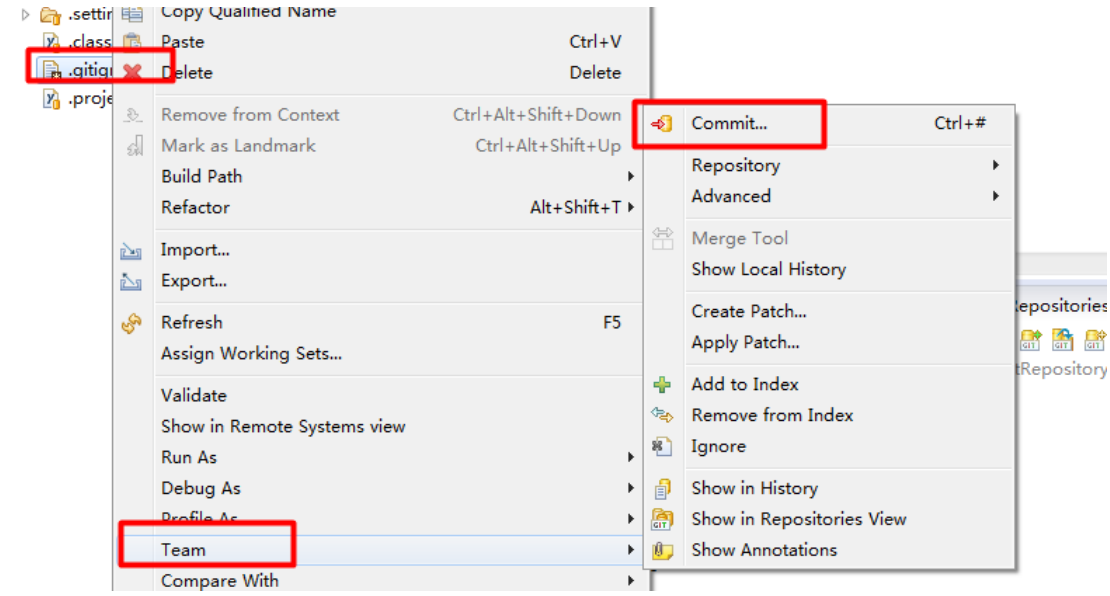
匹配 -*.xxx 、 xxx.* 没有‘/’的匹配，对应的是所有目录中的有效内容。

上述配置中的‘/’代表的是工程的根目录。

修改*.gitignore* 文件后，需要将修改后的文件增加到索引：



提交修改后的*.gitignore* 文件。



保证忽略信息生效。

再次提交相关内容时，会自动忽略*.gitignore* 文件中的配置内容。不做提交处理。