

Eclipse 中 EGit 插件使用--升级版

<http://my.oschina.net/songxinqiang/blog/194203>

注：只参考内容，文档规范按照作业一要求

首先嘞瑟一下，上一篇博客成为了开源中国代码托管的“帮助”教程🙏，有兴趣的可以在空间里面找一下。习惯使用 eclipse 做开发，中途也听别人说其他的 IDE 更好怎么的，去尝试了一下，觉得还是 eclipse 好，毕竟是 IBM 出品的。

以前使用 svn 做代码管理，觉得不错，后来使用了开源中国的 git 代码托管，觉得很赞，但是都是简单的提交、提交，根本没有用其他功能，只是基本一下修改，太大材小用了。有一天突然想尝试一下分支啊、标签啊什么的（据说 git 就是在这些方面比 svn 好的），可是网上资料太少了，而且大多数都是命令行的，特别是分支、标签的推送，找不到操作方法，有也只是命令行，最后实在国外的一个网站上找到（曲折中在 youtube 上看的抓屏）。eclipse 用的人很多吧，用 egit 插件的人也很多吧，可是怎么一搜都会教人安装、提交变更呢，怎么没有更深入的呢，于是我决定写一篇博客，方便后来的人学习 egit 插件使用，于是就有了开头说的那篇博客。但是写的并不是很全面，内容也是文字少、图片多，虽然说按照步骤来还是可以操作了，但是始终觉得不好，而且还有一些方面没有涵盖到，但是又不好改的（成为教程了，万一改得不如从前了呢），所以就单独写这篇了。

好了，铺垫完了，下面开始正题。

1、eclipse 和 egit 版本

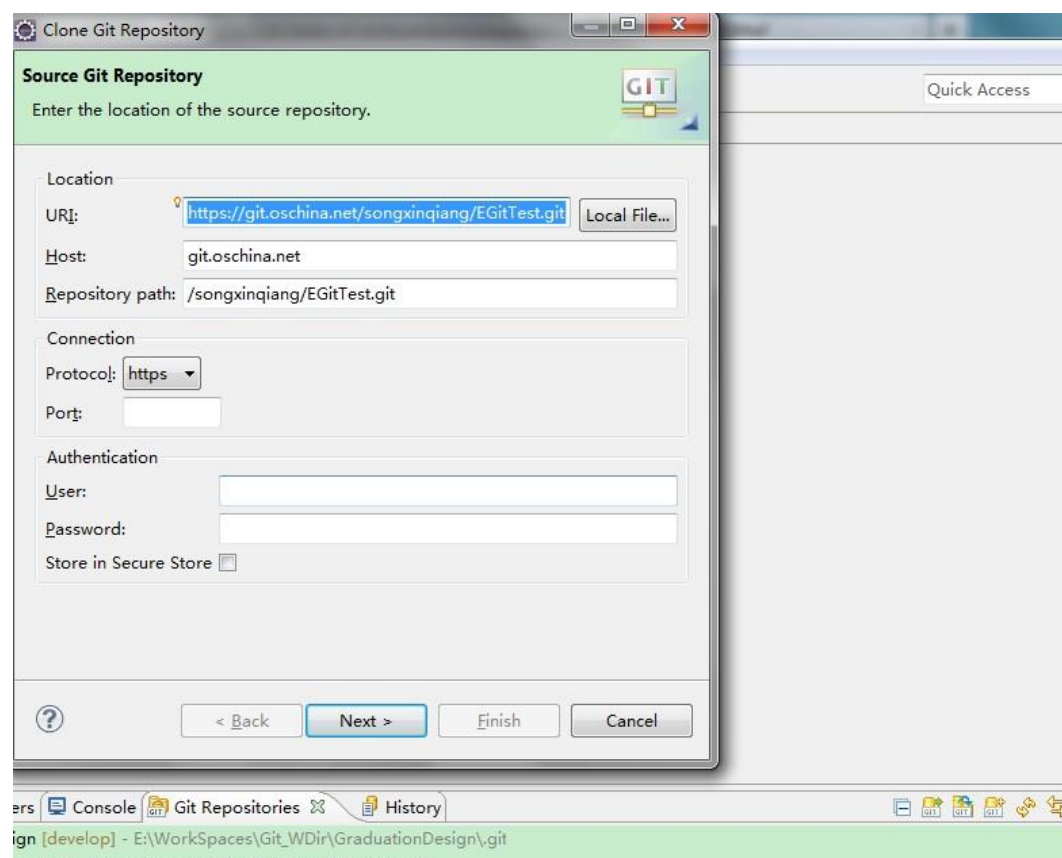
eclipse 使用的是 kepler 的 SR1 版，egit 就是自带的那个版本，要详细的话，我就也只能把文件名贴出来了：eclipse-jee-kepler-SR1-win32.zip。

2、开始 egit 的简单实用

这里也就不说怎么打开 eclipse 了吧，怎么修改配置也就不说了吧，大家看我的 eclipse 和你的 kepler 界面有点不一样，那是我配置了的（这是题外话了），大家可以网上搜一下。egit 插件是自带的，我也就不像其他的文章那样说怎么安装 egit 了，下面介绍怎么通过 egit 简单的使用开源中国的代码托管服务。

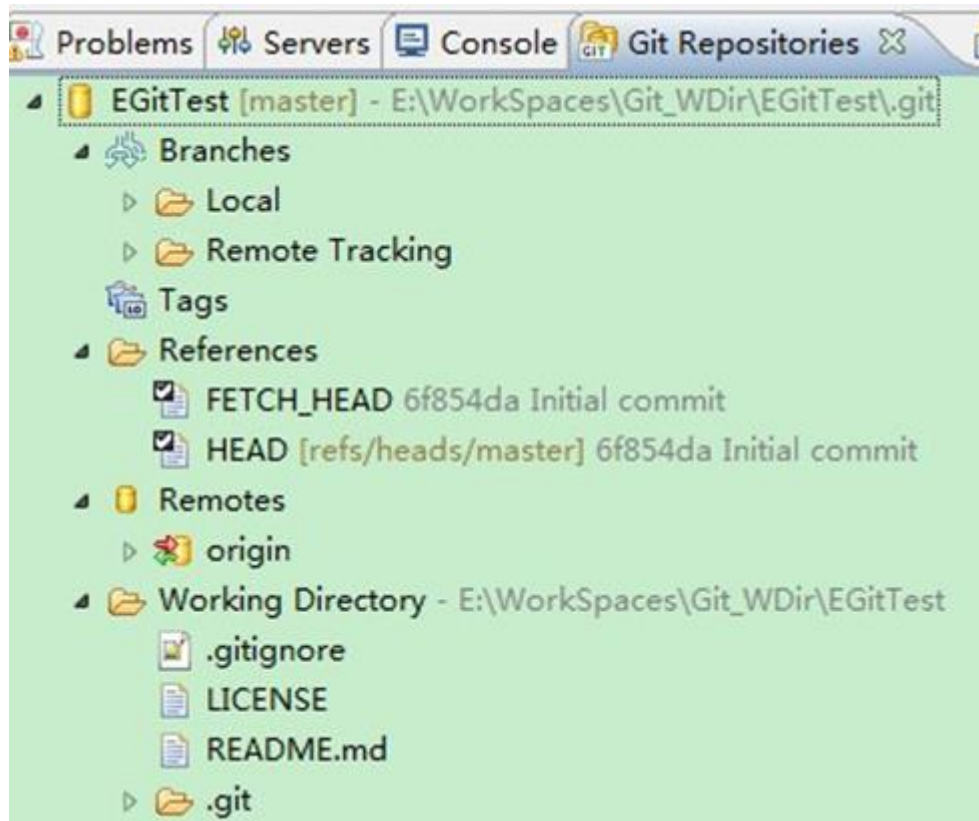
首先你需要在开源中国代码托管建一个工程（或者叫别人把你加入他的工程也可以），反正就是你要有访问一个托管工程的权限。

在 eclipse 中打开名字叫做“Git Repositories”的视图（如何打开？自行百度），然后复制项目的访问地址，在这个视图内 Ctrl+V 就可以进行配置了，或者以也可以找到按钮“Clone a.....” ，



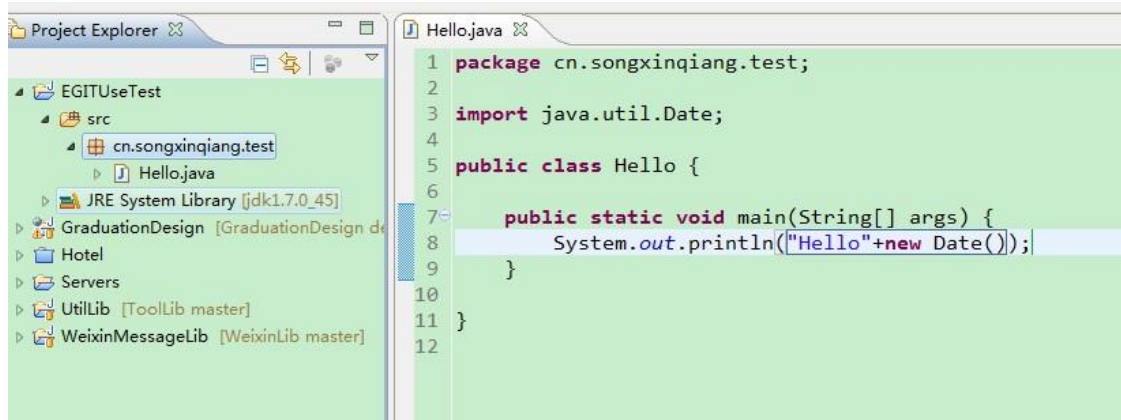
弹出的界面如上，系统已经帮你填写了一些东西，你只需要将你的账号和密码填写好就可以了，接下来你需要的是点击“Next”和“Finish”然后等待（具体多久取决于网速和要下载的量）。

克隆好后效果就是这样的：

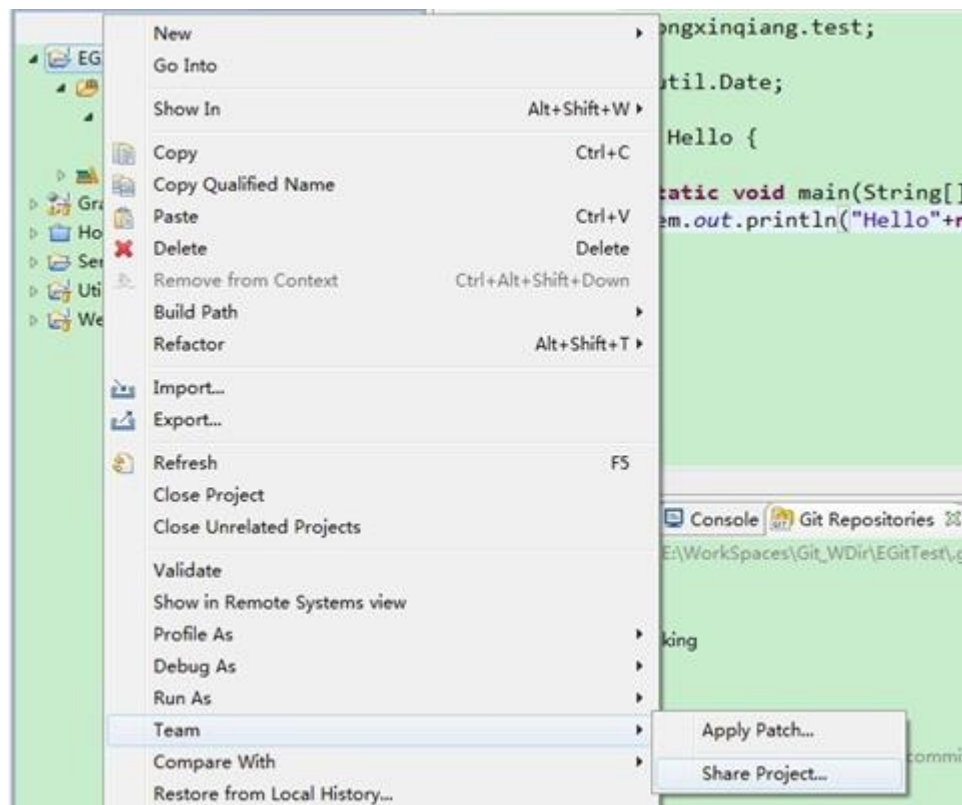


我的是空的项目，如果你的项目已经有代码的项目应该还要多一个文件夹，里面放着有各种托管文件。

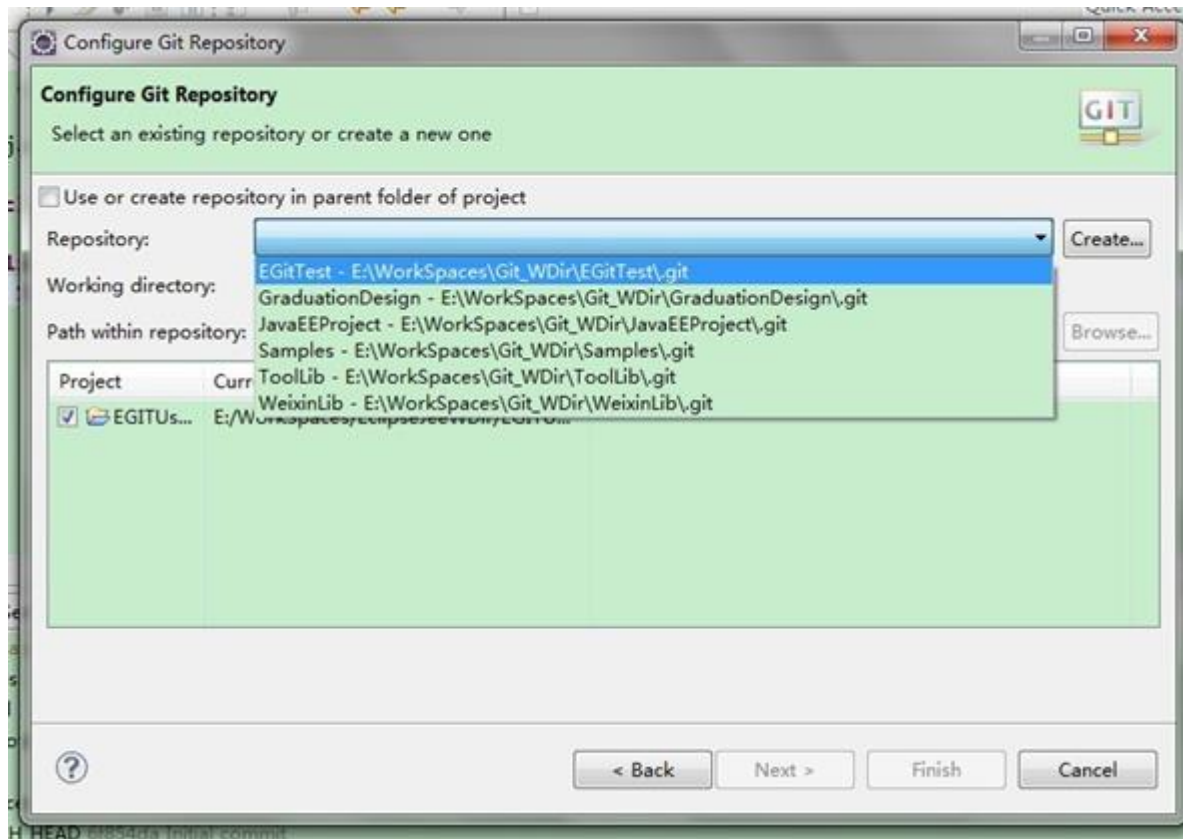
接下来为了演示，按照常规建一个 java 工程并且添加一个类，结果就像这样：



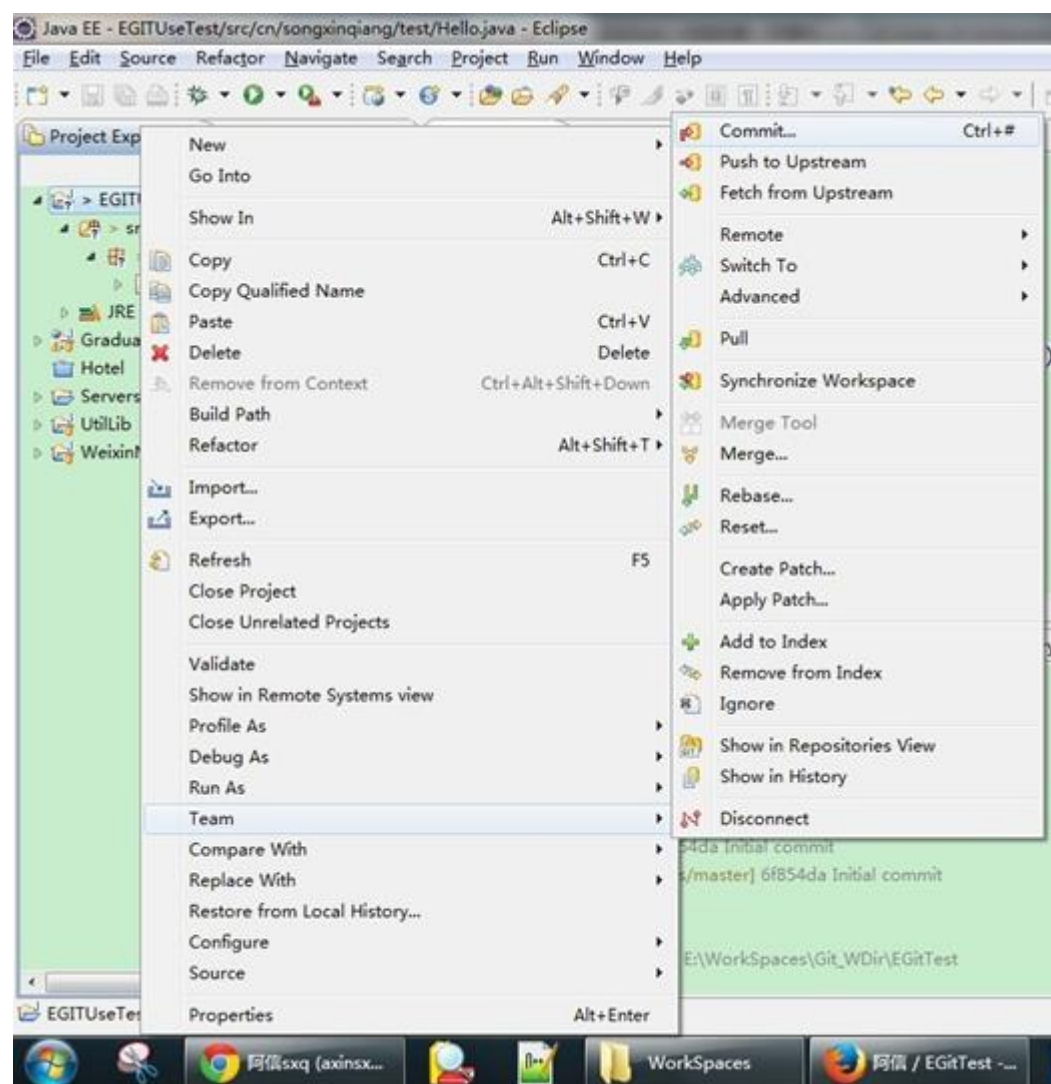
然后一步一步的将这个项目托管到 git 中去：在项目上右键，选择 “Team->Share Project”



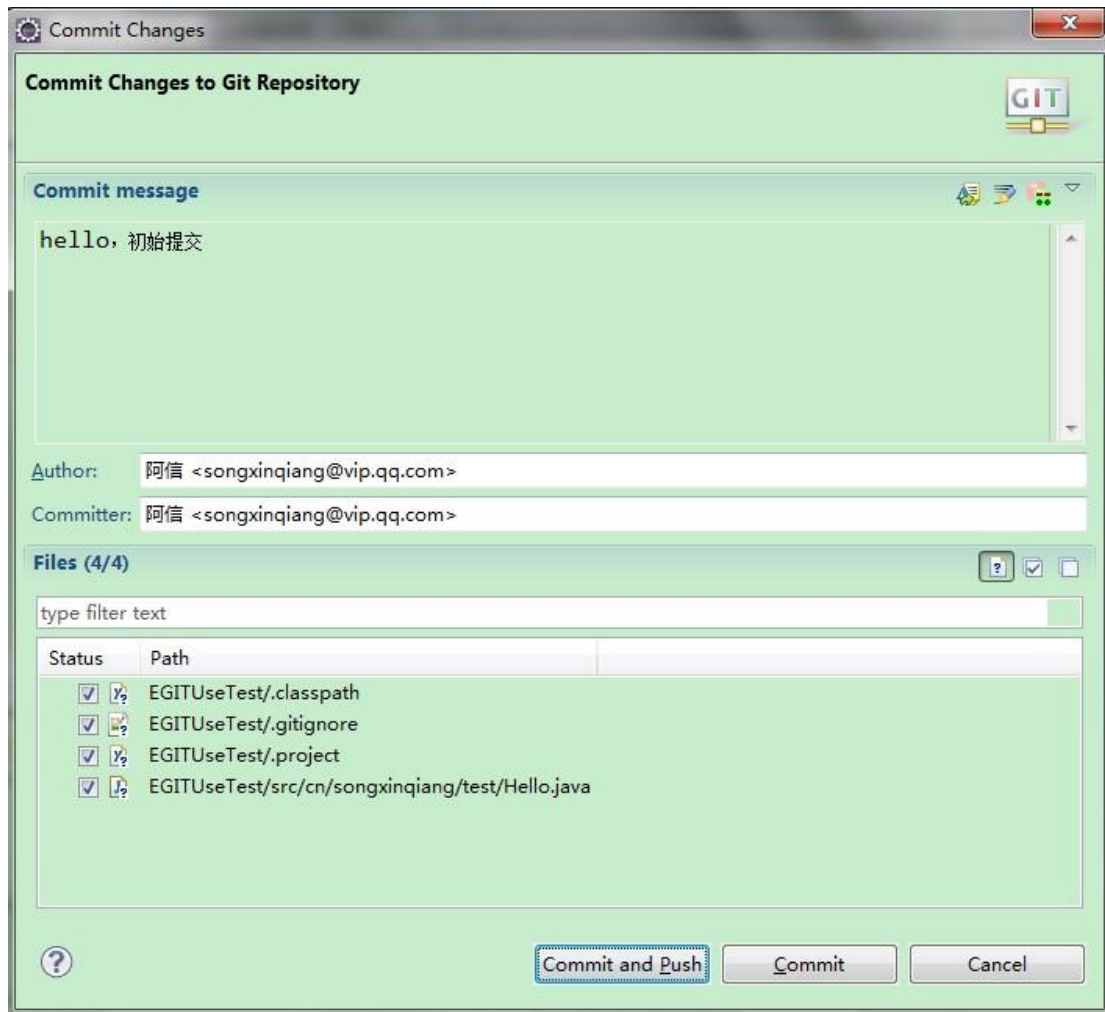
然后选择 git，next，然后选择刚才克隆的代码库（如果只有一个不需要选），



最后点击“Finish”就可以了，接下来就是要将代码提交到代码库中去，在项目上右键，选择“Team-->Commit”



然后填写好注释勾选所有的文件，点击提交并推送就行了

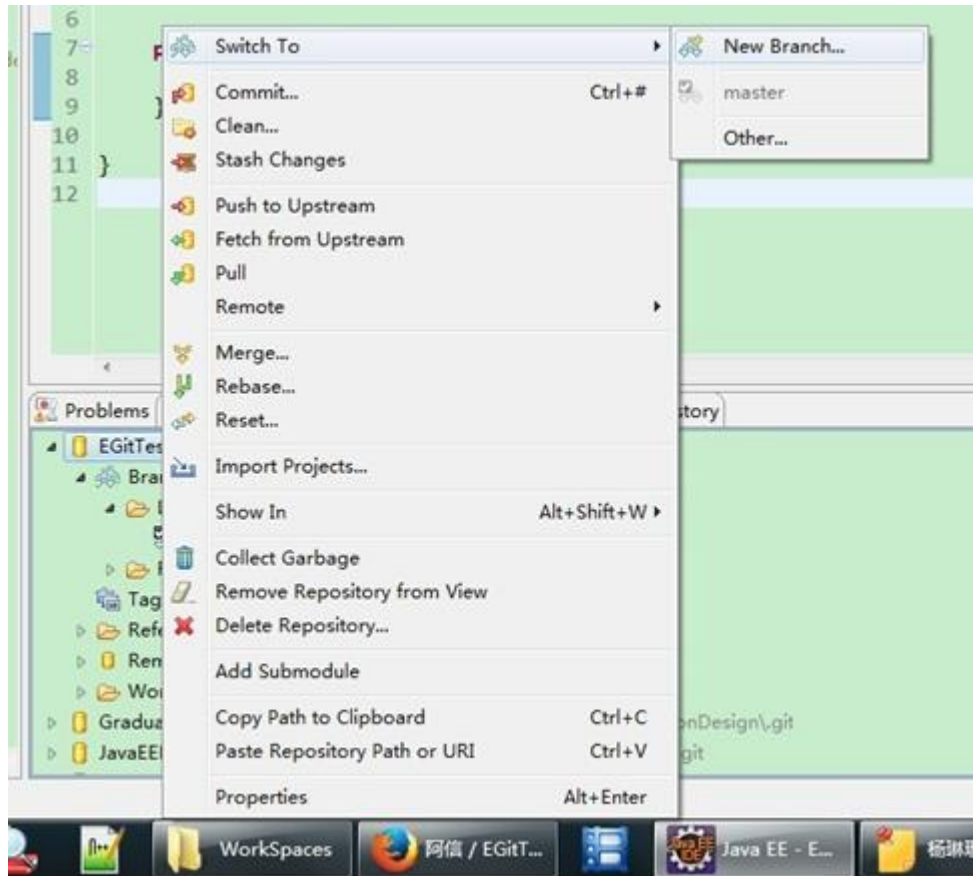


基本的使用就是这些，不断地写代码、提交代码、推送代码，当然这是一个人开发，用 git 只是记录一下代码，以后万一需要可以找回，也就是我以前的用法（现在也没好到哪里去），但是有一点大材小用了吧，接下来说一些“高级”的。

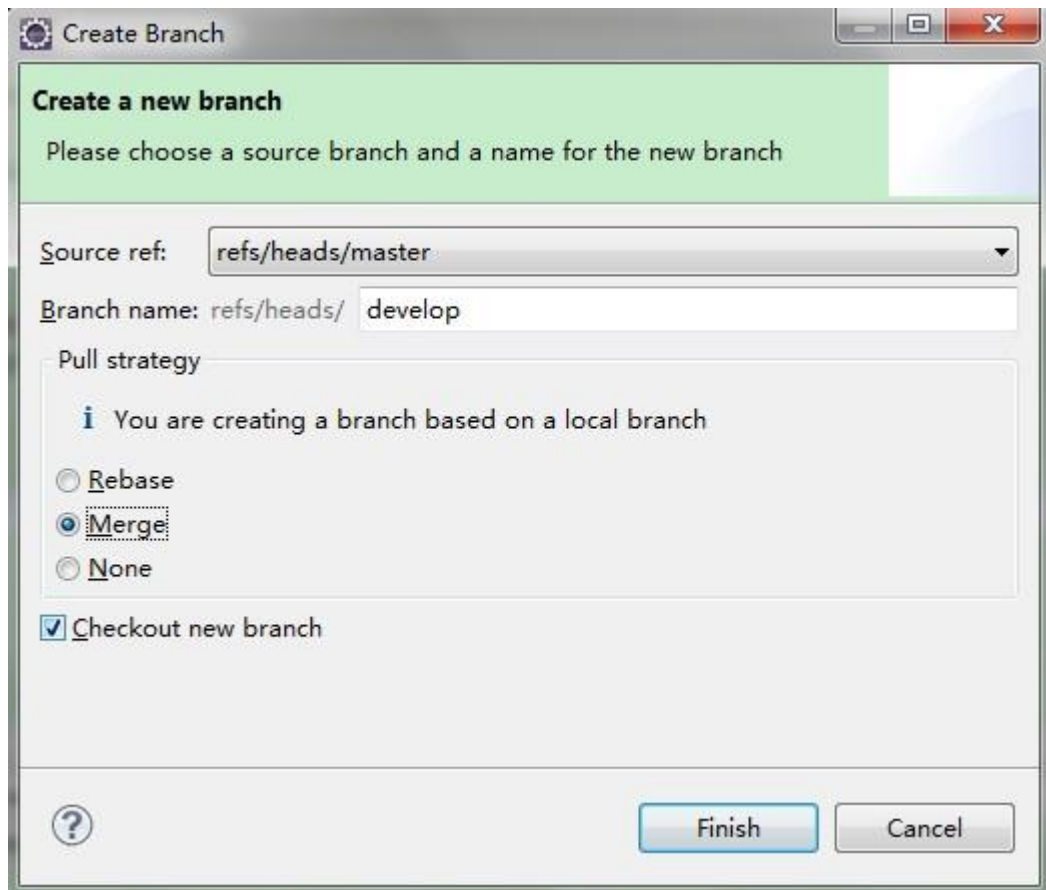
3、分支（创建、推送、使用）

分支的好处、分支的作用、为什么要分支？这些都不是我想说的（我想说也说不清，还是让那些大师去说吧），要想知道的话，请自行上网搜索哦，我接下来要说的是如何通过 egit 创建、使用、推送分支（删除的话由于是后面截的图就放在后面来说）。

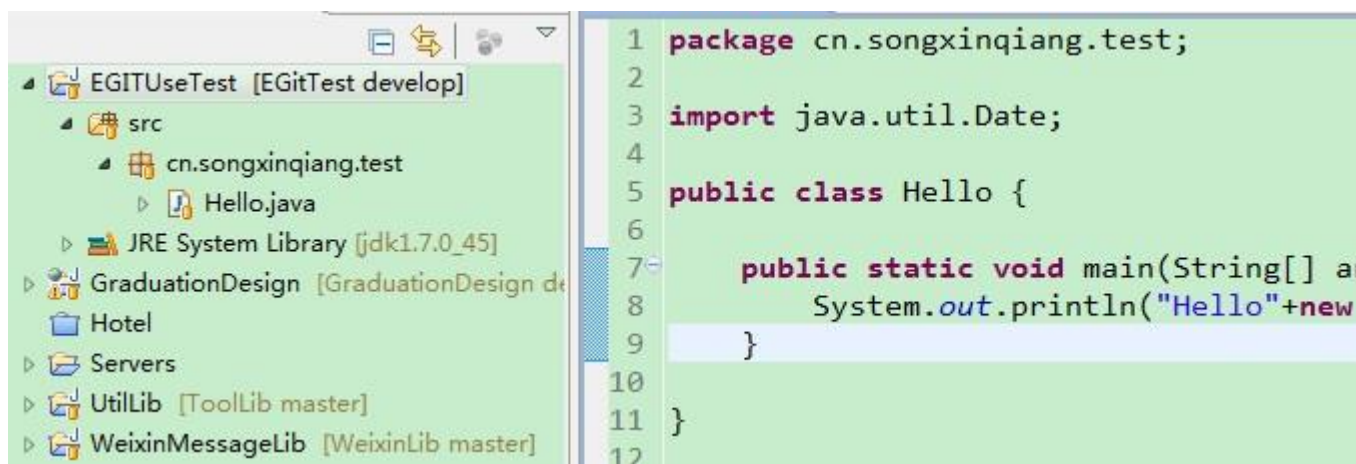
回到代码库的视图，在关于代码库的那一行上面右击，“Switch to--> new”（如果以后有了分支直接选就是了）：



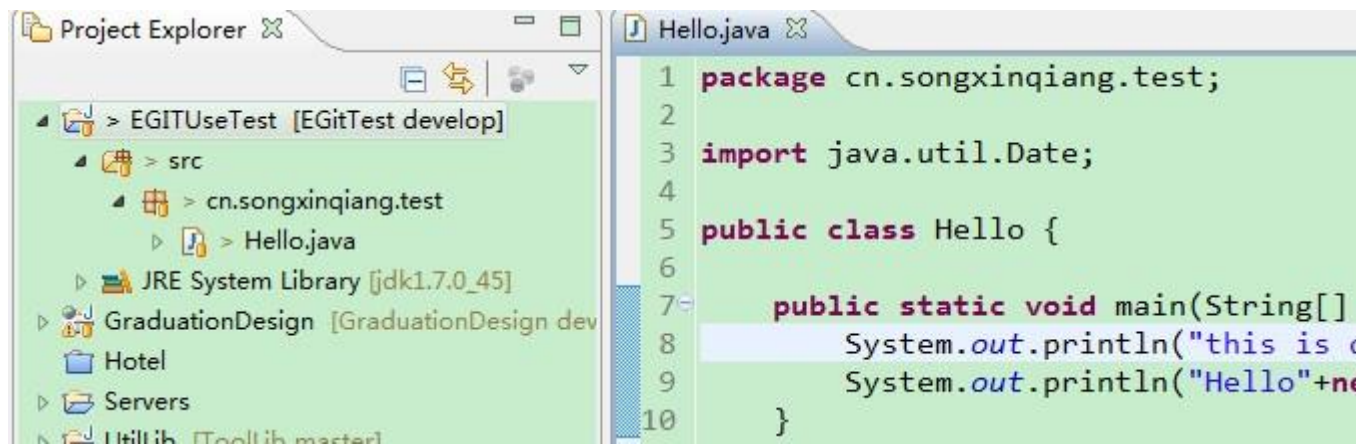
然后填写分支的名称，从哪里来，用什么方式来等，然后点击“finish”就可以了：



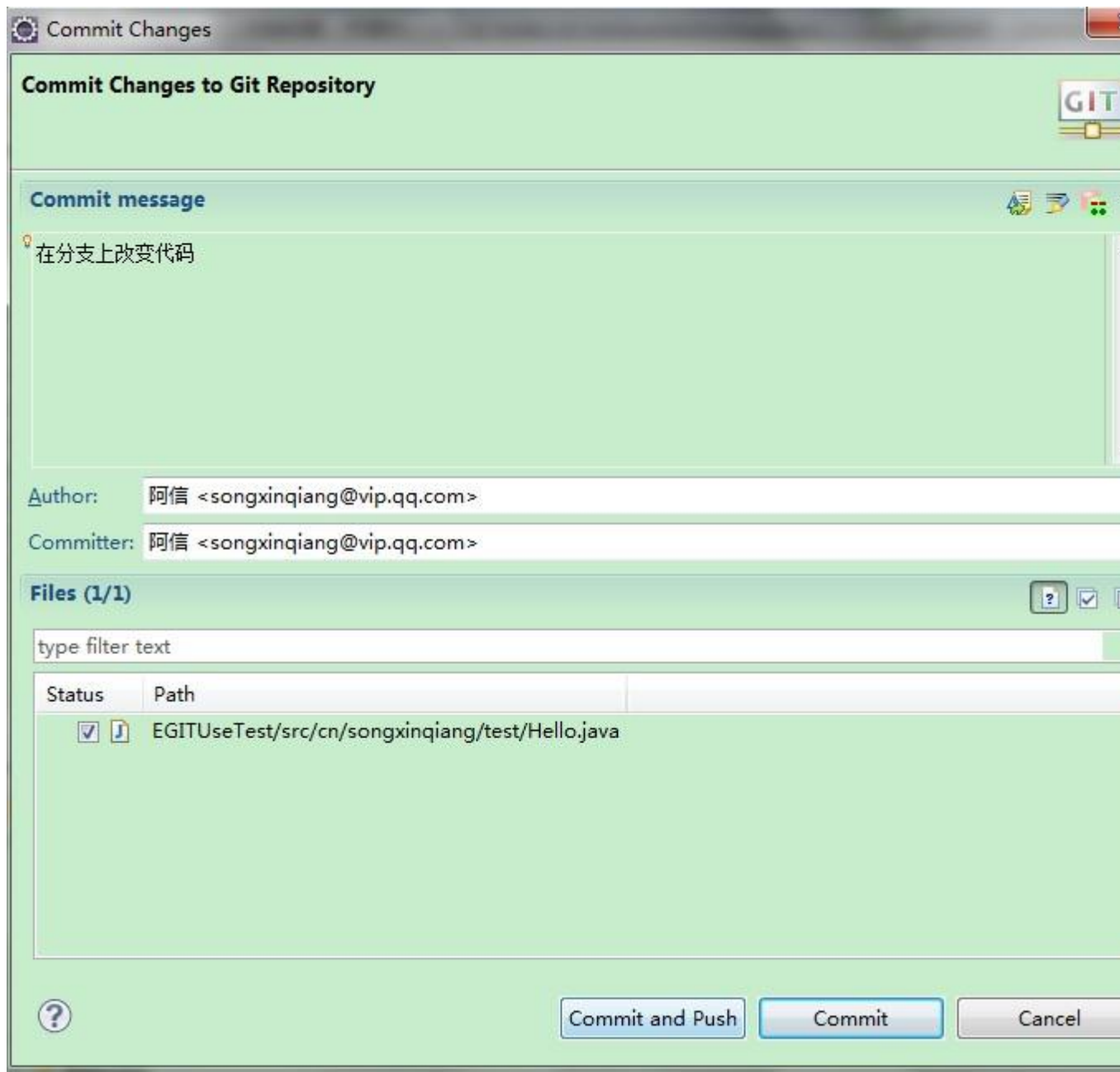
由于勾选了“checkout.....”所以分支建好之后我们已经在使用新的分支了，



接下来改动代码，添加一行打印，



提交改变，填写说明：



这里我直接提交并推送，但是后来才想起还没有说怎么直接推送新分支呢（我刚才的操作已经推送了新分支），下面来补充说一下怎么推送先新建的分支，在代码库的视图上右键“remote-->push”然后 next，出现下图的界面

Push to: origin

Push Ref Specifications

Select refs to push.

Add create/update specification

Source ref: Destination ref:

Add delete ref specification

Remote ref to delete:

Add predefined specification

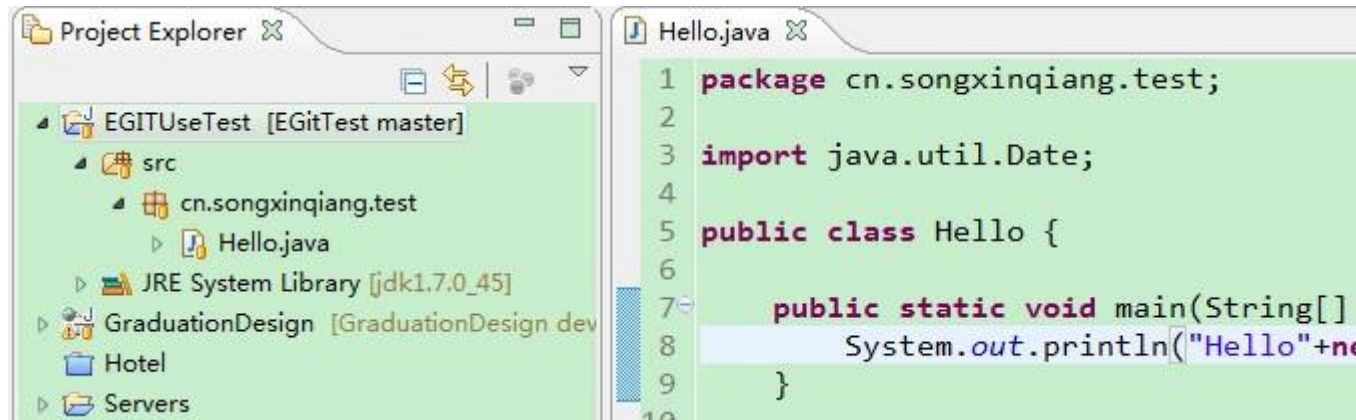
Specifications for push

Mode	Source Ref	Destination Ref	Force Update	Remove
+ Update	refs/heads/*	refs/heads/*	<input type="checkbox"/>	

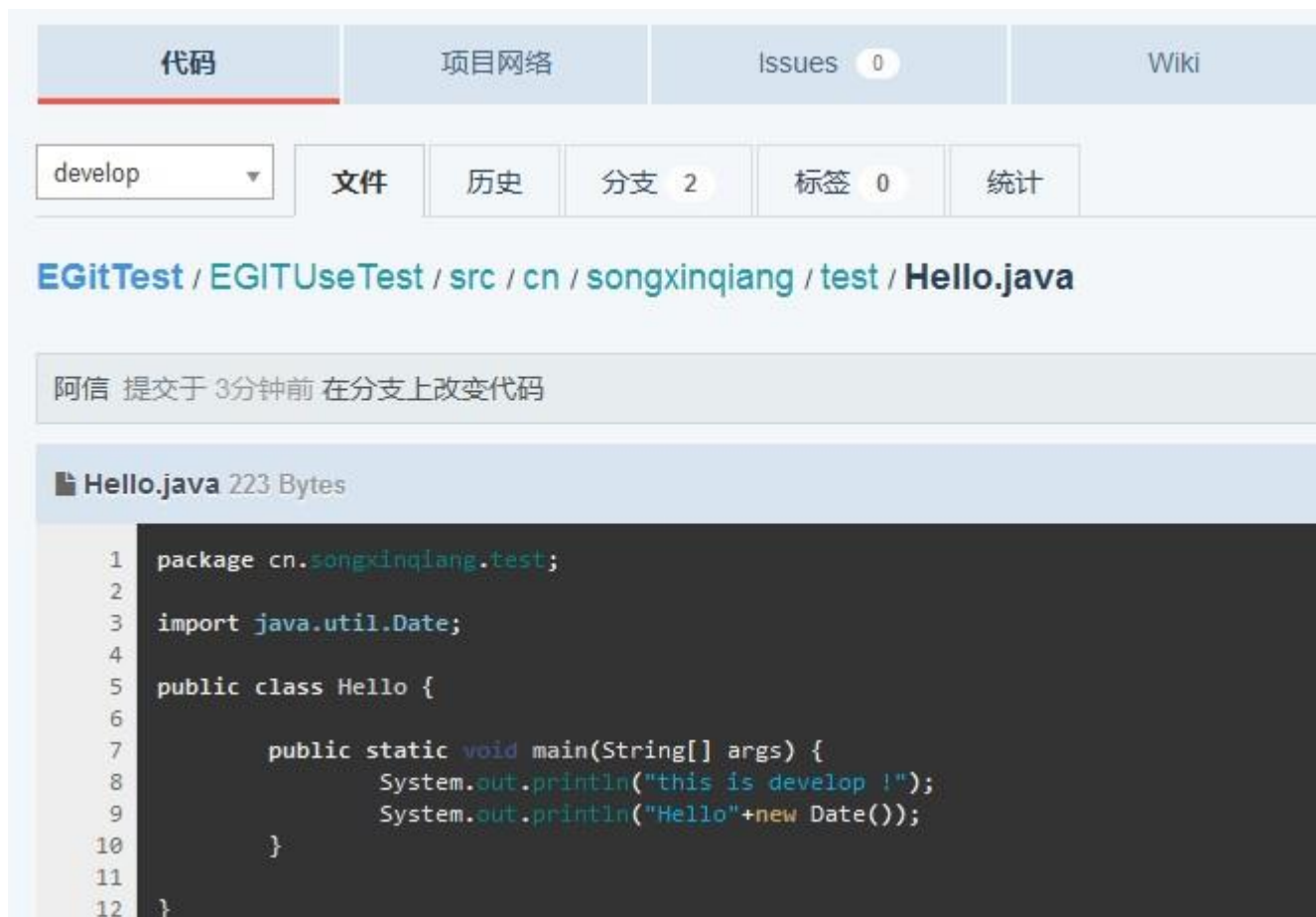
☐ Save specifications in 'origin' configuration

单击“add all branches.....”就会出现中间的那个“update.....”了，然后 finish 就会把所有的分支都更新（新分支会推送出去），当然这里偷懒了。

我们在 develop 分支上更改了代码，在 master 分支上是没有变化的，切换过去看一下：



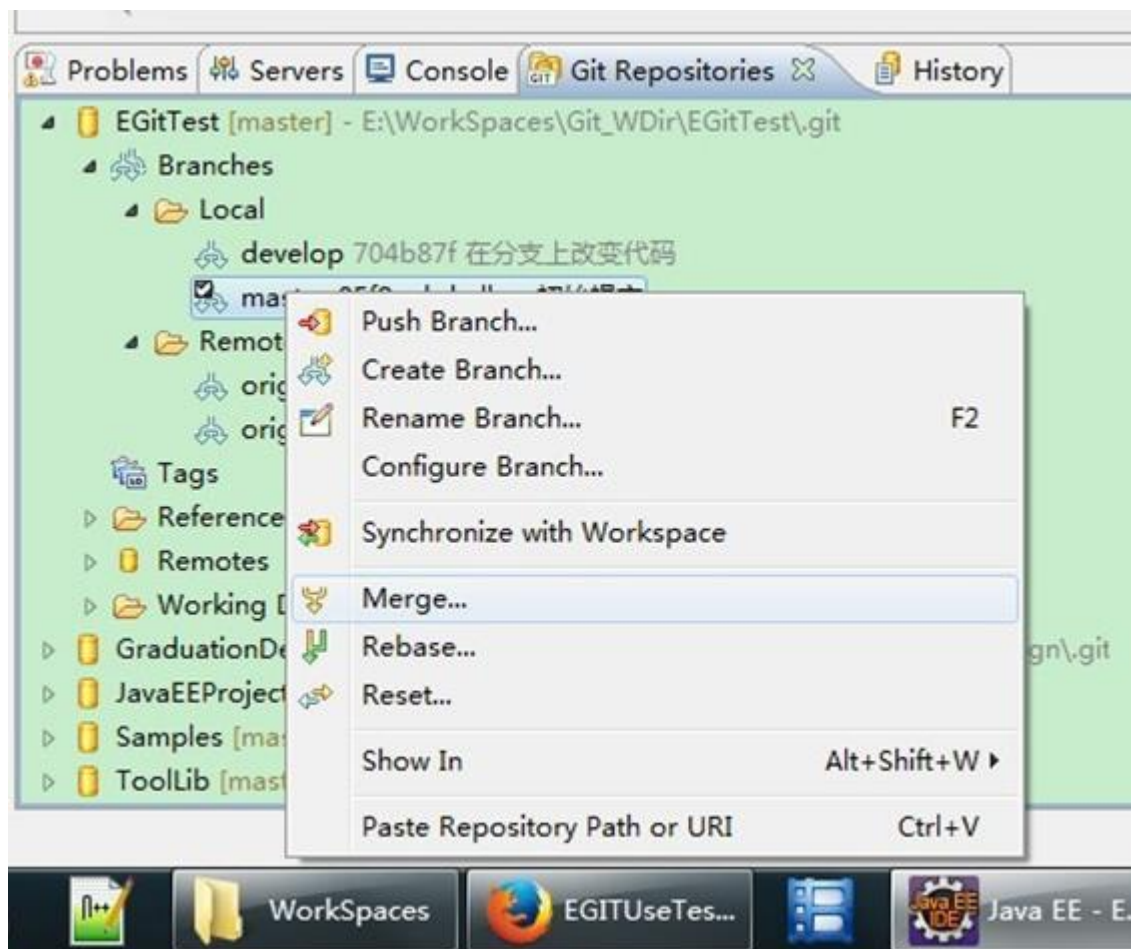
服务器呢，分支 develop 变化了，master 也是没有变化的：



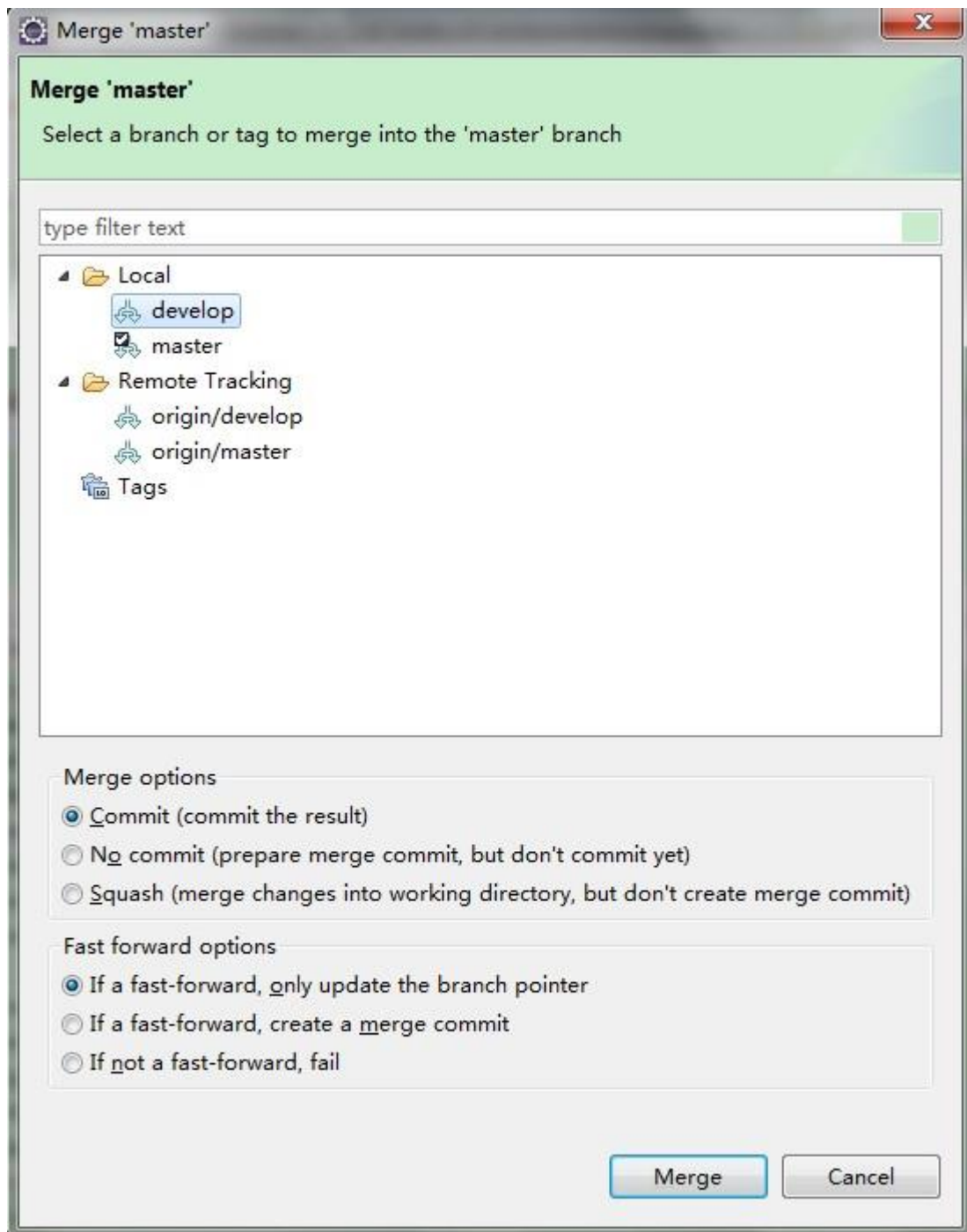


这样两个分支的代码就不一样了，接下来我们要将分支 develop 的代码整合到 master 分支中去，这也是正常的开发中要做的，在一个分支中开发（大项目甚至更多），开发好后或者主干上有重大变动时，进行代码合并。

要将 develop 的代码合并进 master，需要首先切换分支到 master，然后再代码库视图上在本地的 master 上右键“merge”

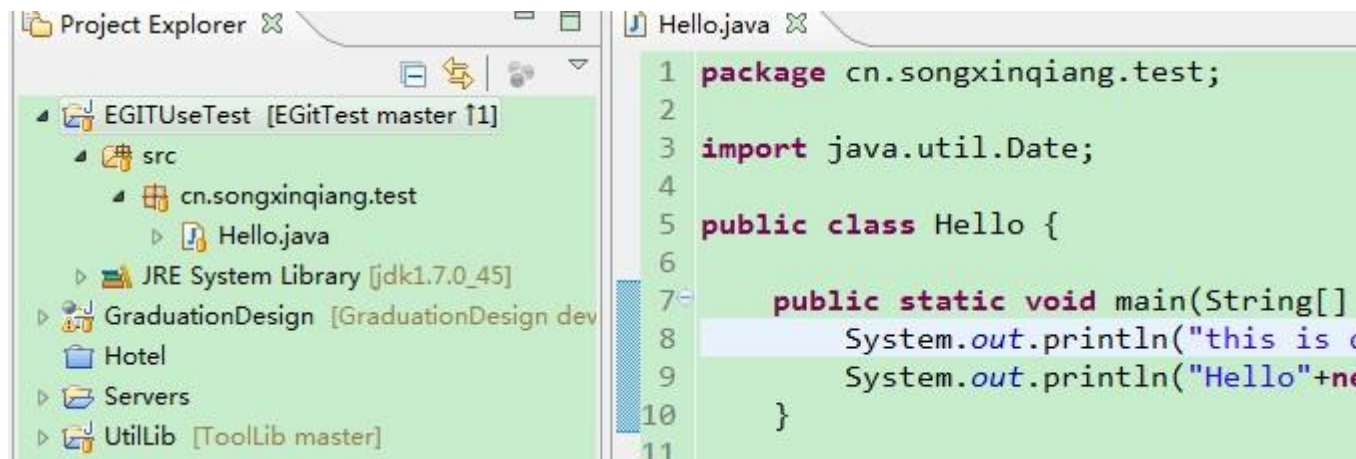


然后在弹出的界面中选择要合并进 master 的分支：develop：

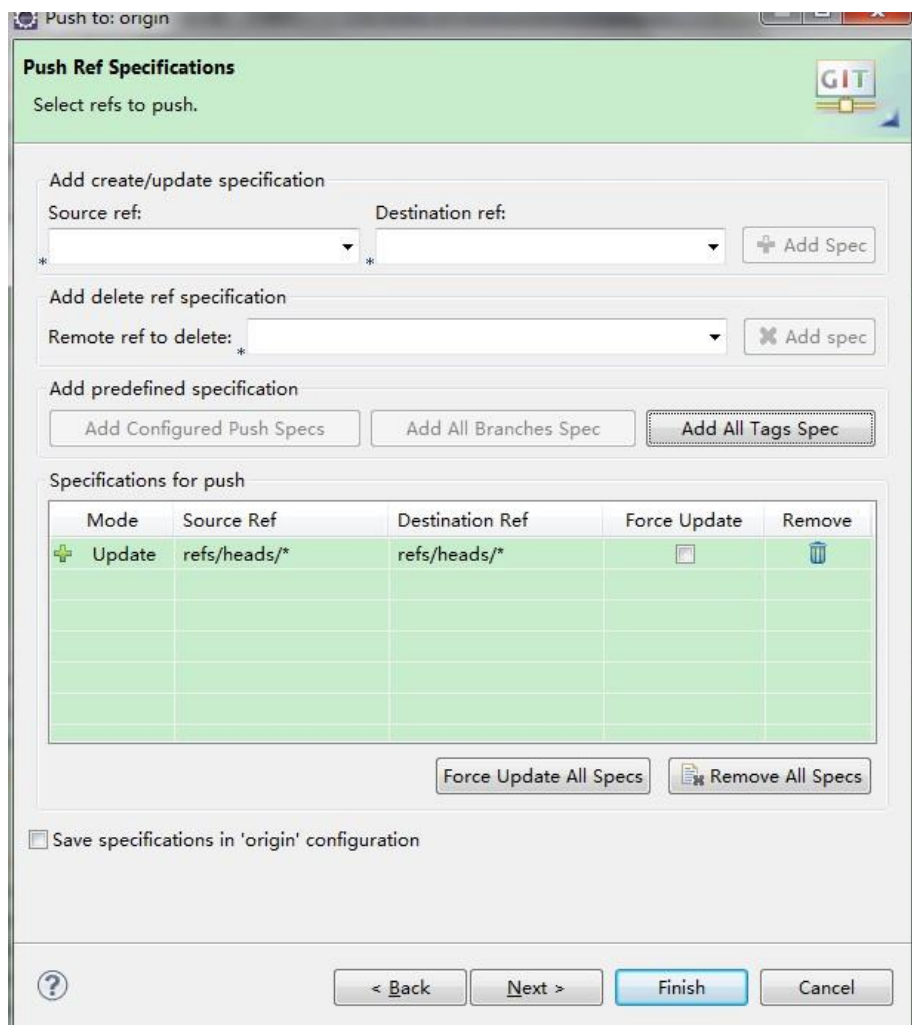


点击“merge”，如果有冲突的话会在代码上有标记，到时候需要解决之后再继续，

合并之后的效果：

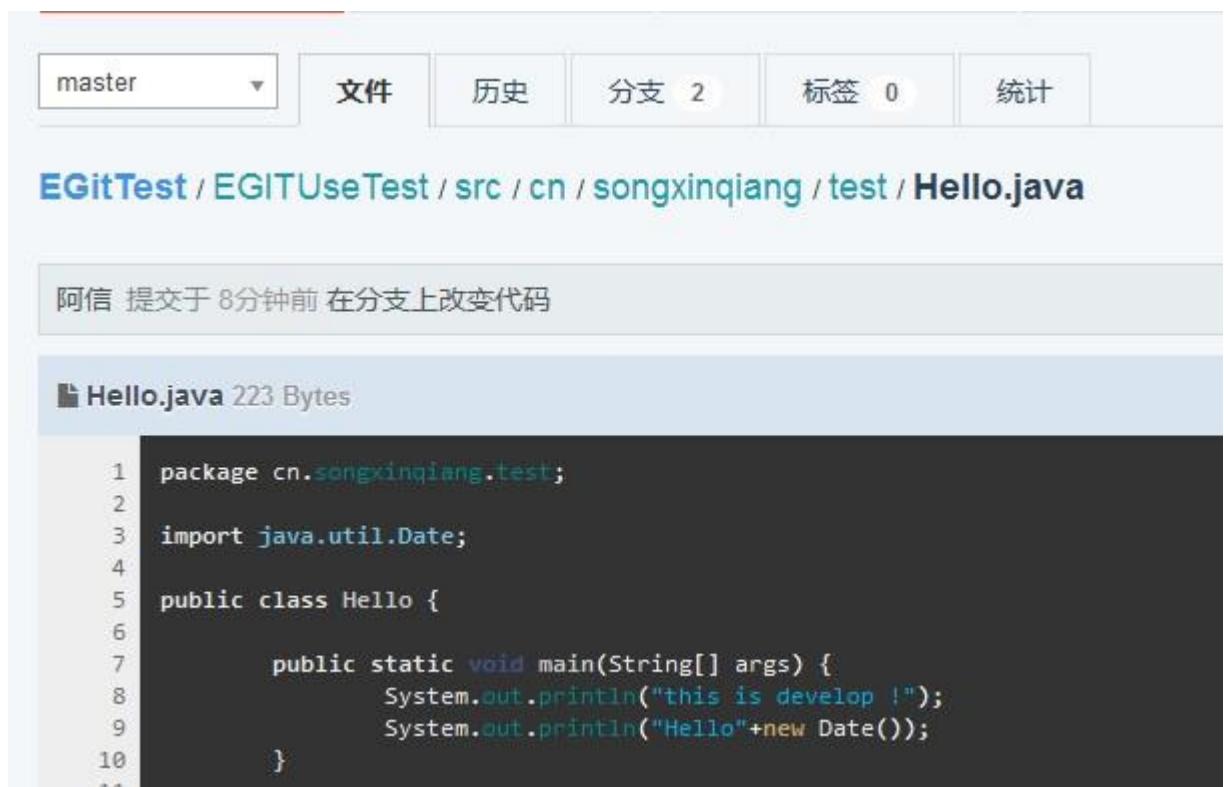


可以看到我添加的那一条打印语句已经合并进来了，由于还没有推送，所以在项目上有一个“11”，那么接下来我就推送到服务器上去。在代码库的视图上右键，“remote- -> push”，然后选择“add all branches.....”，点击 finish





然后在服务器上就可以看到 master 分支的代码已经更新了：

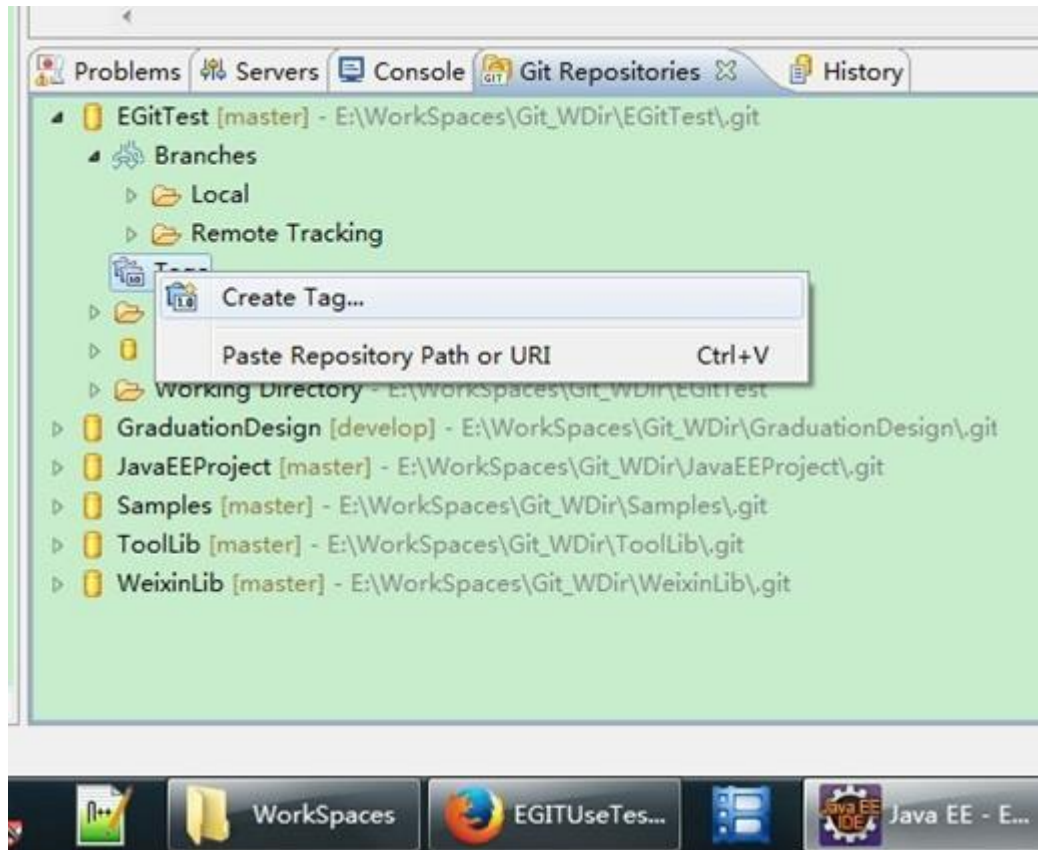


分支的使用就是这些了，是不是很简单、很快捷？我们在开发中对于合并分支、提交关键代码更改这样的大事是不是需要记录一下，虽然可以看历史记录，但是还有一个更方便的东西----标签，下面就介绍如何使用标签。

4、标签（创建、推送）

如前面所说，标签就是用来记录大事件的，便于以后查看，标签不止一种，但是我不介绍（网上有的是，而且我也说不明白），需要的可以自行上网搜索。

创建标签很简单，只需在代码库的视图的“Tags”上右键，“Create tag”，



弹出的窗口中，填写标签的名称、说明、基于哪一版代码，然后 finish：

Tag name*:
test-tag1

Tag message*:
这个是标签的说明信息，可以方便查看

☐ Force replace existing tag

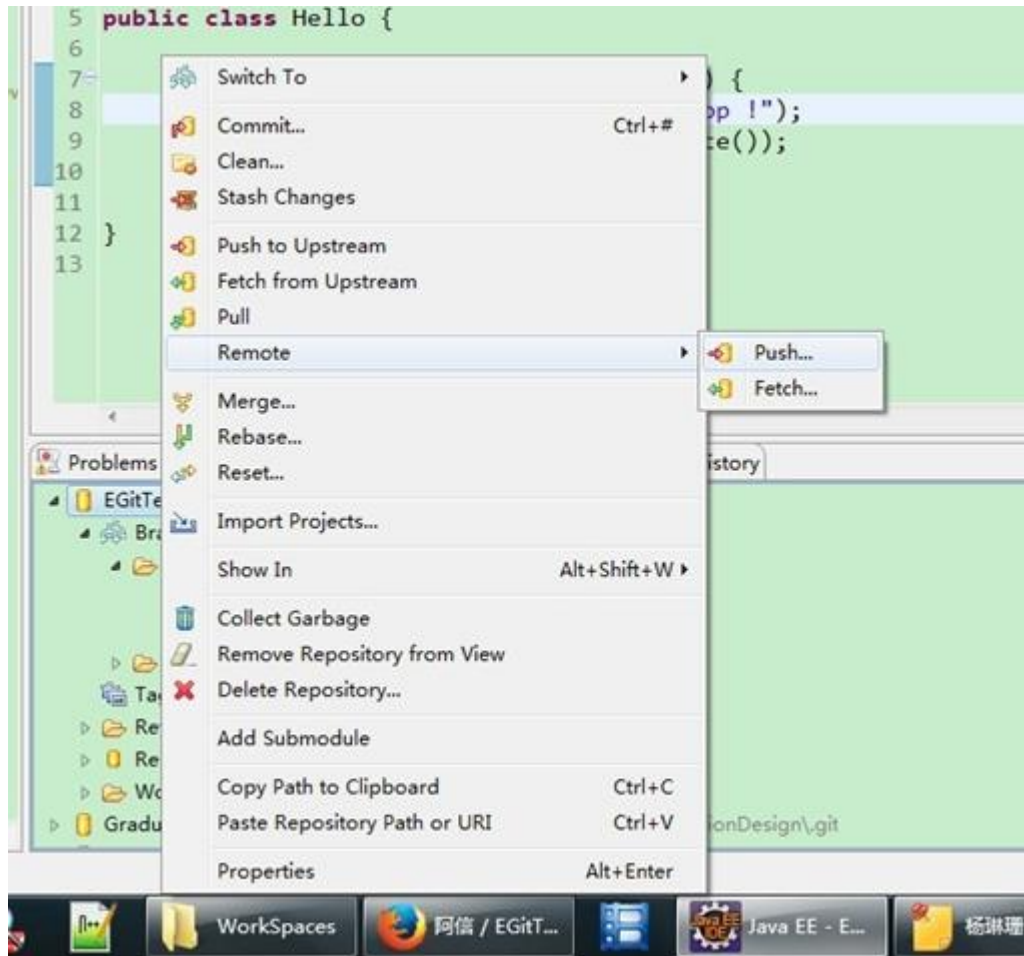
▼ Advanced

Choose commit that should be associated with this tag.

704b87f1: 在分支上改变代码

Clear

接下来要做的是将标签推送到服务器，方法也是在代码库的视图上右键 “remote-->push”




借来点击一次 next，然后点击“add all tag.....”推送所有标签更改，当然和分支一样，新标签也会推送出去

Push to: origin

Push Ref Specifications

Select refs to push.



Add create/update specification


Source ref: Destination ref:

Add delete ref specification

Remote ref to delete:

Add predefined specification

Specifications for push

Mode	Source Ref	Destination Ref	Force Update	Remove
+ Update	refs/tags/*	refs/tags/*	<input type="checkbox"/>	

☐ Save specifications in 'origin' configuration

标签使用就介绍到这里了。

5、分支、标签的删除



分支在完成该分支的开发后可能要删除（不知道最佳做法是删除还是不删除），把标签有时候需要删除（标签原则上不删除）。反正就是有时候要删除，下面介绍一下删除的做法。

为了演示删除标签，特意新建一个标签，在服务器上查看的情况是这样的：

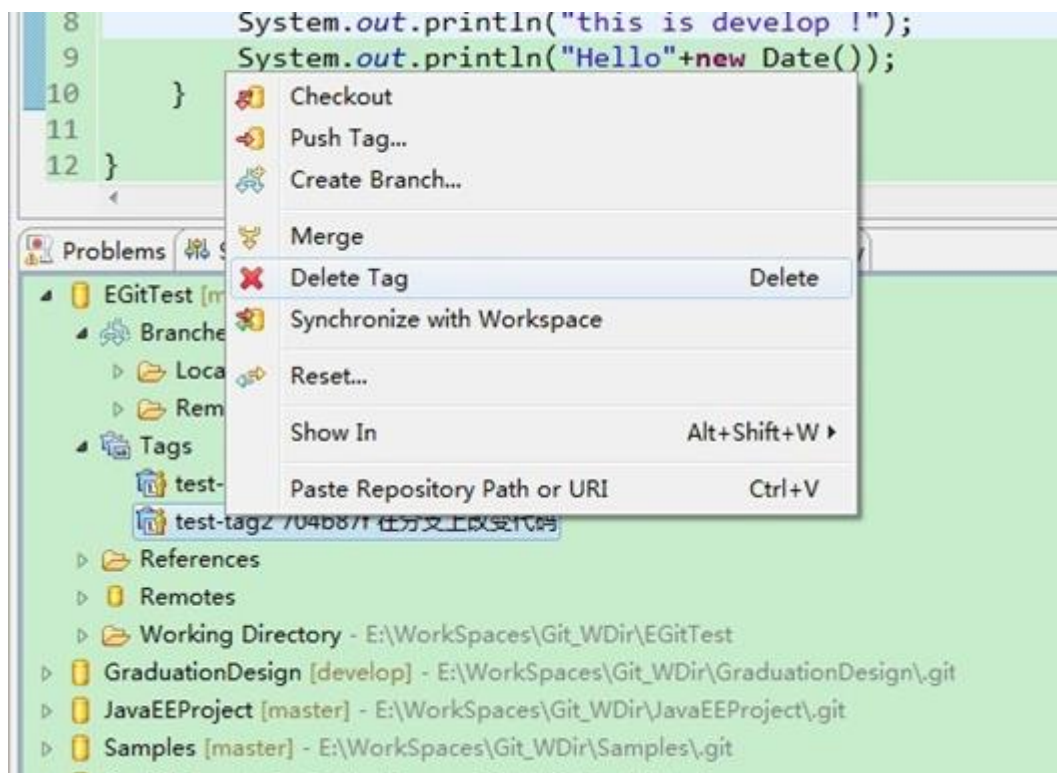


删除标签，需要在代码库的视图上右键“remote-->push”，然后“add all tags.....”添加一栏，可以看出来标签在远程的记录是“refs/tags/*”我们将前面的“update”单击一下变成“delete”，悲剧出现了：后面的列都不见了，没办法手动填写吧（分支还好可以辅助生成，这个只能手动，可能是标签真的不应该删除吧），先前我们知道标签在远程的记录是“refs/tags/*”，我们旧照样子写，然后八角具体的标签名写上，就成了这个样子：

Specifications for push

Mode	Source Ref	Destination Ref	Force Update	Remove
 Delete		refs/tags/test-tag2	<input type="checkbox"/>	

然后点击 finish 远程的标签就删除了，当然为了防止下一次将这个标签推送出去，我们需要在本地也删除这个标签，在标签上右键“delete”：



这样服务器上的标签就没有了：

阿信 / EGitTest

代码项目网络Issues0Wiki

master文件历史分支2标签1统计

 **test-tag1** 这个是标签的说明信息，可以方便查看

704b87f1e – 在分支上改变代码

再次提醒，标签一般不删除。

删除分支也差不多，为了演示我也建了一个叫做 delete 的分支，

阿信 / EGitTest

代码项目网络Issues0Wiki

master文件历史分支3标签1统计

最新创建

所有分支

分支名	最后提交
 develop	 704b87f1eaf 在分支上改变代码 1
 delete	 704b87f1eaf 在分支上改变代码 1
 master 默认	 704b87f1eaf 在分支上改变代码 1

删除步骤就是在代码库的视图上右键，“remote-->push”然后 next，然后可以使用向导生成删除请求，不用像标签那样手动写了，

Push to: origin

Push Ref Specifications

Select refs to push.

Add create/update specification

Source ref: * Destination ref: *

+ Add Spec

Add delete ref specification

Remote ref to delete: *

X 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/delete	<input type="checkbox"/>	

Force Update All Specs Remove All Specs

☐ Save specifications in 'origin' configuration

? < Back Next > Finish Cancel

方法为：上面的上面的“Source ref:”和“Destination ref”选择要删除分支，然后点击后面的“add spec”，这个时候下面记录的是“update”我们点击一下这个 update 就会变成“delete”了，然后点击 finish 远程服务器上的分支就会被删除，



可以看到，服务器上已经有了 delete 分支，当然本地的 delete 分支也最好一并删除了，方法和删除标签一样，就不截图了。

6、使用远程已有本地还没有的分支

在远程服务器上别人推送提交了一个新的分支，或者本地的代码库是刚克隆的而原来已有不止一个分支，那么本地是不会有除 master（默认分支）以外的分支的（remote 下面有），那么如何得到呢？

本人尝试过在 remote 下面的分支上 checkout，但是代码出来了，更改推送不出去。那么在代码库的视图上“fetch from.....”呢，这样根本就没有提示有版本变化！

经过不断尝试、探索，关键在于在 local 下面没有分支的记录，最后终于找到，在新建分支的窗口上，选择来源就用 remote 的那个分支就行，截图就像下面：

Create Branch

Create a new branch

Please choose a source branch and a name for the new branch

Source ref: refs/remotes/origin/develop

Branch name: refs/heads/ develop

Pull strategy

☐ Rebase

☒ Merge

☐ None

☒ Checkout new branch

?

Finish

Cancel