**Git**

|  |
| --- |
| 最易懂的教程,也是笔记。（两个网址内容一样）：<https://www.zhihu.com/question/41667536/answer/486640083> http://www.cnblogs.com/tugenhua0707/p/4050072.html   1. git常用命令：   http://www.ruanyifeng.com/blog/2015/12/git-cheat-sheet.html 2.Git教程-菜鸟教程 **<http://www.runoob.com/git/git-tutorial.html>**  **可以学习和查询git的相关命令。** |

**Git是必备技能。**

|  |
| --- |
| 慕课网学习笔记：   1. Git是什么？ 2. Git和svn的区别 3. 安装git   直接去官网下载，安装就一直next到install。   1. 初始化用户名和邮箱 2. 创建版本库 3. 提交、查看状态 4. 查看提交日志，版本回退（回退到上一个、上n个、指定版本） 5. 理解工作区和暂存区 6. 撤销修改和删除文件git checkout -- readme.txt（--前后都有空格） 7. 创建公钥 8. Push到github 9. 从github克隆下来 10. 创建和合并分支 11. Github多人协作 |

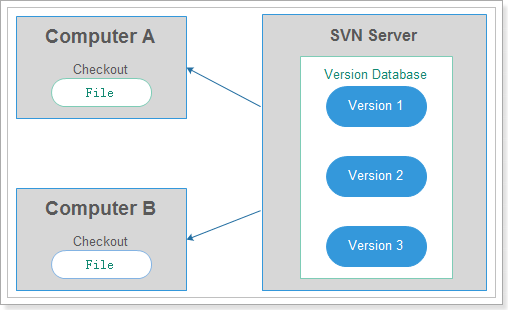
|  |
| --- |
| Git命令分以下几个模块（帮助记忆）：  1.新建代码库（init）  2. 配置（config）  3.增加/删除/重命名文件（add、rm、mv）  4.代码提交（commit）  5.分支（branch、checkout）  6.标签（tag）  7.查看信息（log、status、diff、show。。）  8.远程同步（remote、pull、push）  9.撤销（reset、stash、checkout） |

# Git与Svn对比

## Svn

**集中式**版本管理控制系统（版本库是集中放在**中央服务器**的），

下图就是标准的集中式版本控制工具管理方式：



集中式版本控制工具缺点很明显：

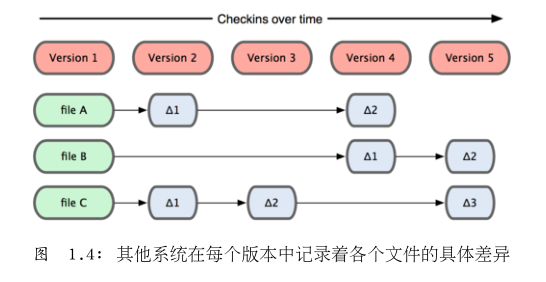
1. 中央服务器故障，则历史版本就丢失了,而开发者个人电脑上只存在一个历史版本，即**单点故障问题**。

## Git

Git是**分布式**版本控制系统，每个人的本地电脑都有存储历史版本，避免了单点故障问题。但常规来说，还是会有一个远程仓库。

### Svn记录具体差异

Git和其他版本控制系统的主要差别在于，Git只关心文件数据的整体是否发生变化，而多数其他系统则只关心文件内容的具体差异。



### 三种状态

现在请注意，接下来要讲的概念非常重要。

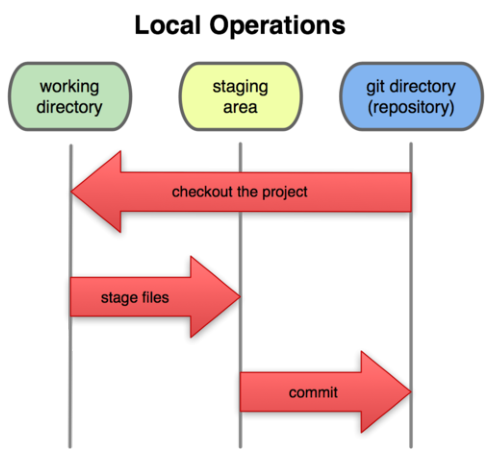
对于任何一个文件，在 Git 内都只有三种状态：已提交（committed），已修改（modified）和已暂存（staged）。

已提交：表示该文件已经被安全地保存在本地数据库中了；

已修改：表示修改了某个文件，但还没有提交保存；

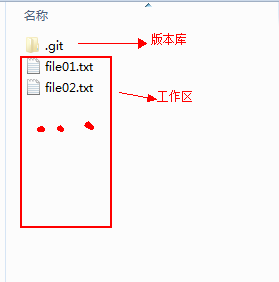
已暂存：表示把已修改的文件放在下次提交时要保存的清单中。

由此我们看到 Git 管理项目时，文件流转的三个工作区域：Git 的本地数据目录，工作目录以及暂存区域。

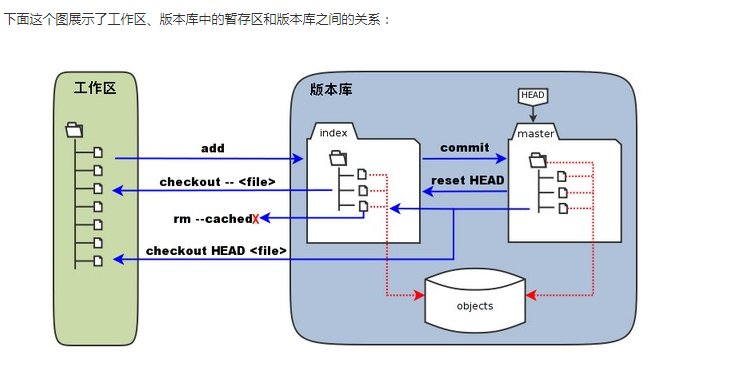


1. 如果是 git 目录中保存着的特定版本文件，就属于已提交状态；
2. 如果作了修改并已放入暂存区域，就属于已暂存状态；
3. 如果自上次取出后，作了修改但还没有放到暂存区域，就是已修改状态。

### 三种区域



**暂存区：**英文叫stage, 或index。一般存放在 ".git目录下" 下的index文件（.git/index）中，所以也叫作索引（index）。暂存区是在版本库中。



### Git的代码托管中心

* 局域网下：

- Gitlab服务器

* 外网下

- Github

- 码云

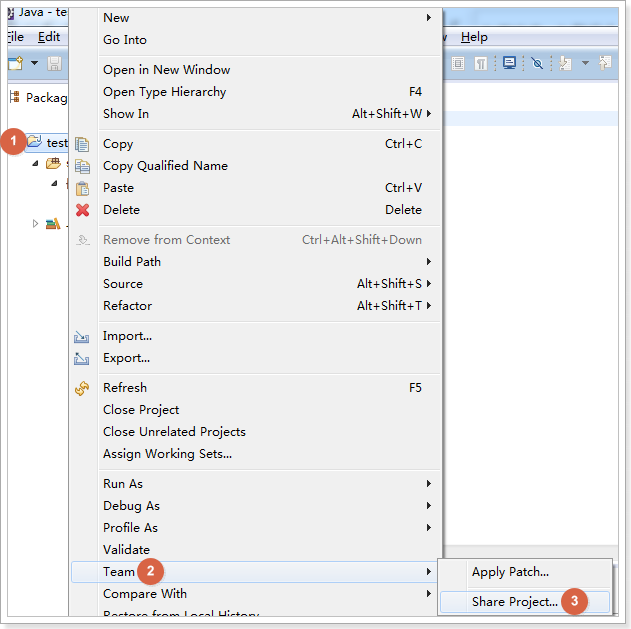
**代码托管中心的作用：维护远程库**

# Eclipse的Git使用

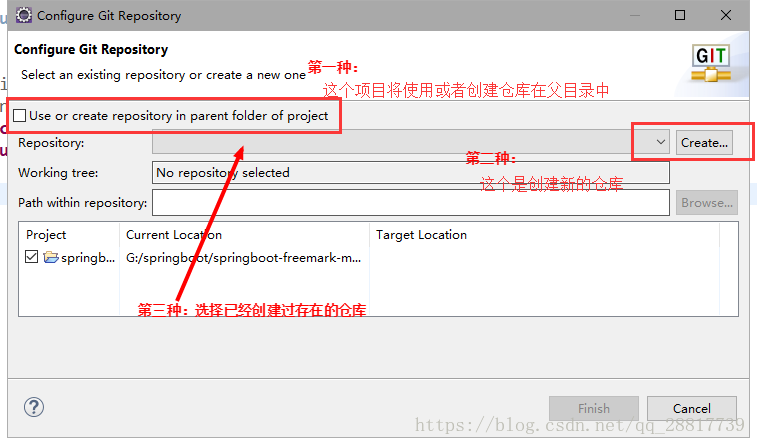
https://zhuanlan.zhihu.com/p/51479929

## 提交代码

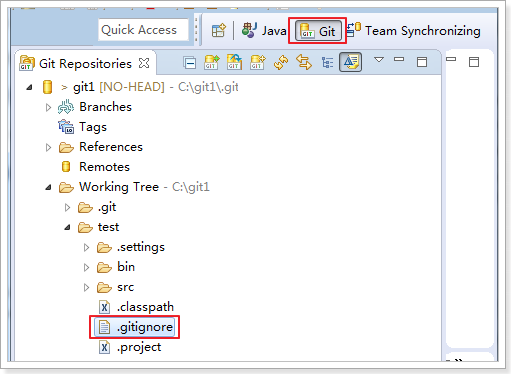
### 设置使用Git



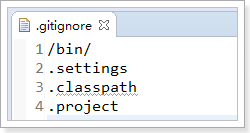
### 创建git本地仓库



### 设置忽略的文件



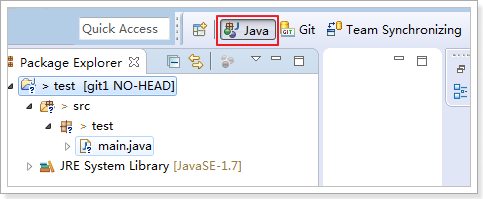
如下图，加入检入时需要忽略的文件



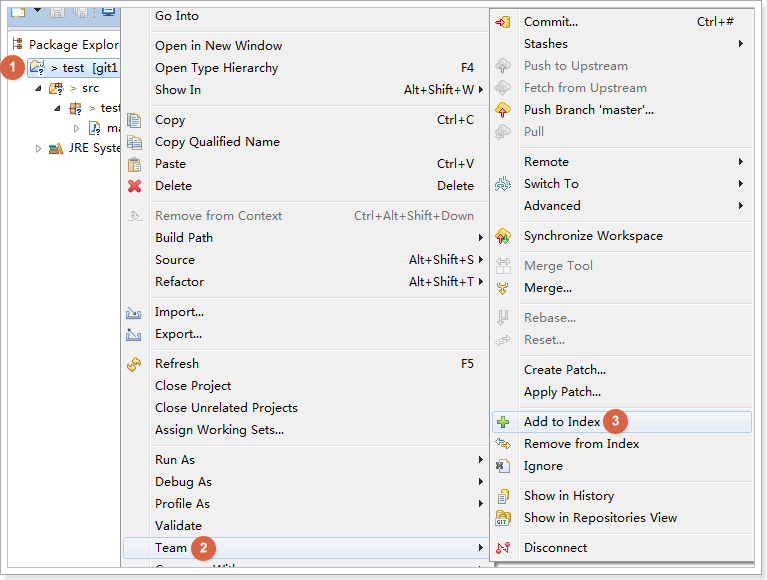
/target/\*\*等等

### 提交代码到本地(暂存区)

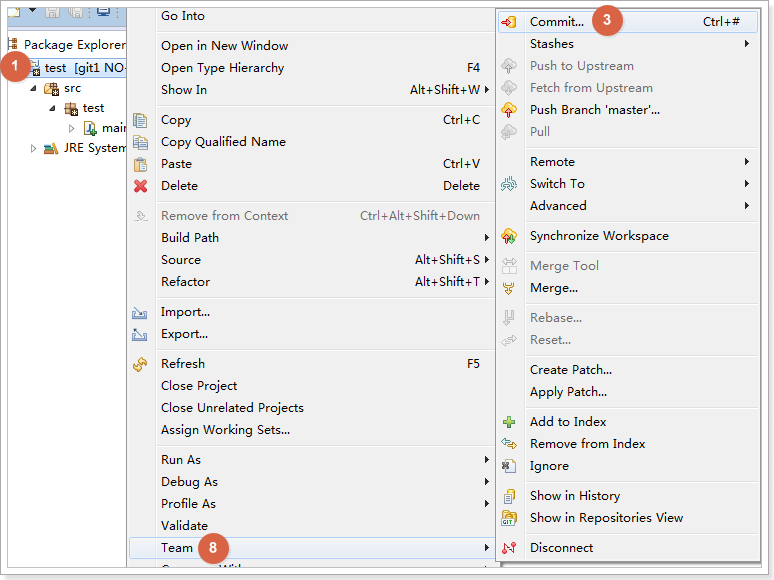
回到java视图

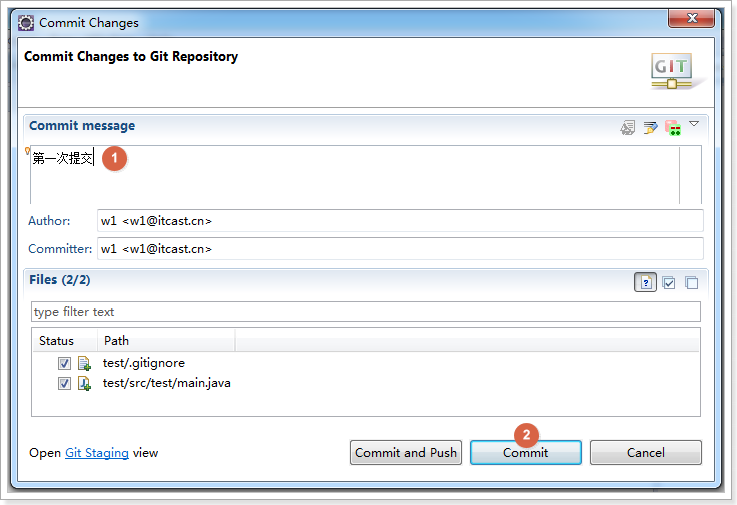


添加代码到暂存区中（Myeclipse有便捷一些的操作界面）



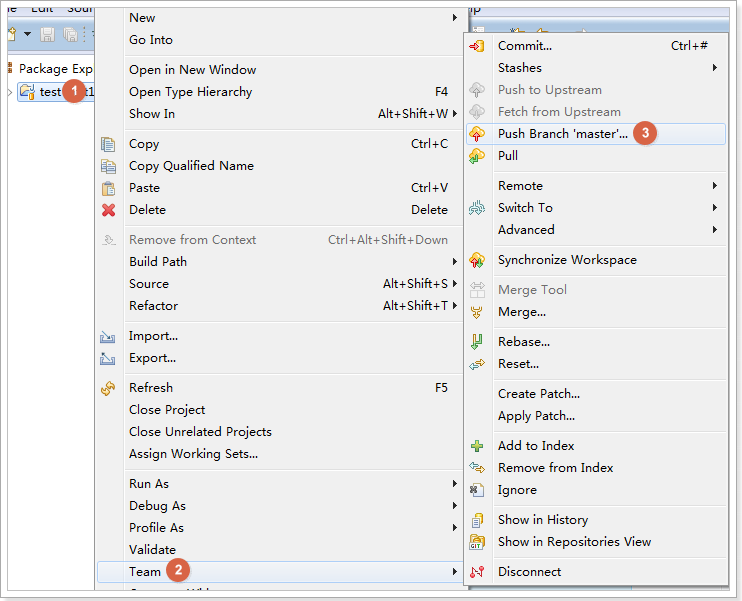
提交代码到本地仓库



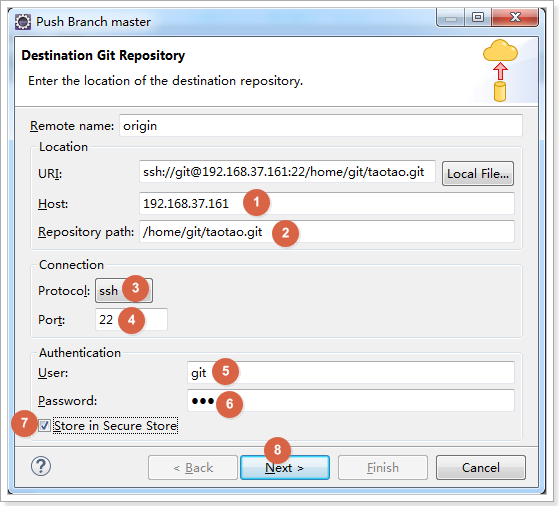


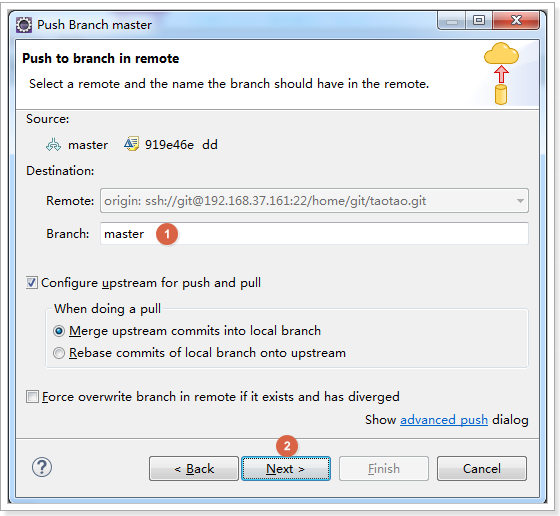
### 提交代码到Git服务器

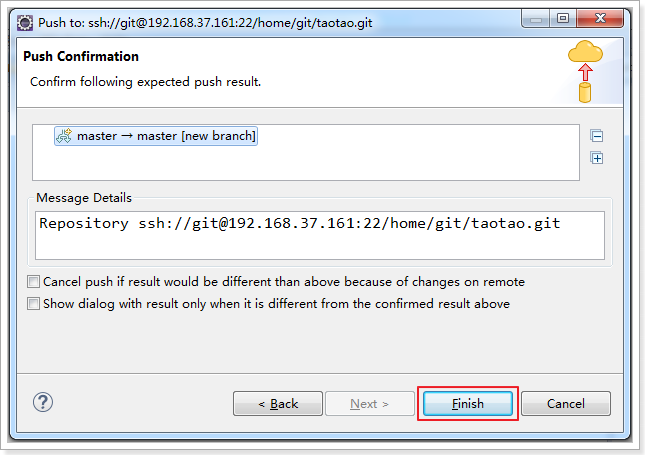
这里的git服务器是安装在linux系统上面的。



设置git服务器地址：

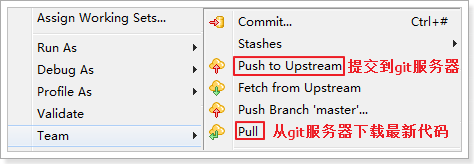






### 检入检出git服务器操作

当第一次提交到git服务器后，后续的代码检入检出git服务器的操作如下图即可。



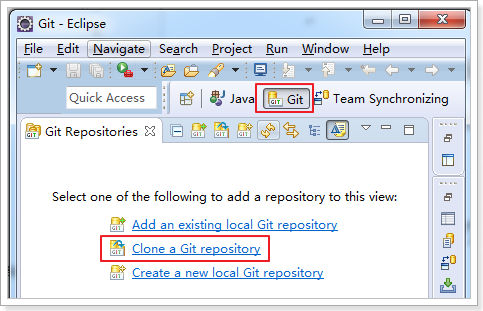
注意：**Commit提交的作用是把代码提交到本地。**

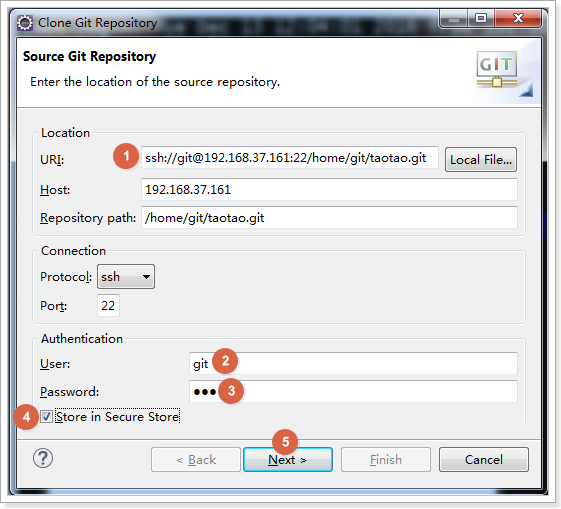
**Push的作用是把本地的代码提交到远程的git服务器**。（并且先要提交到本地仓库之后才可以提交到远程服务器）

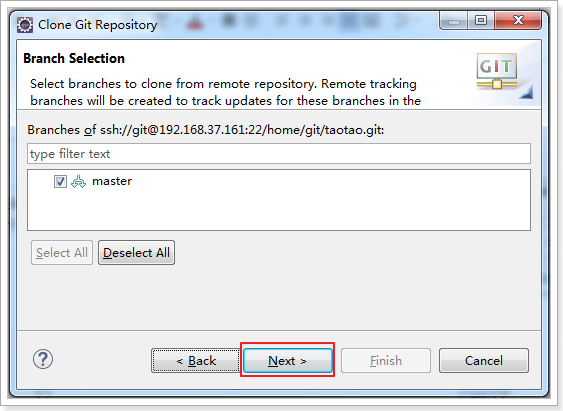
## 检出代码

### 克隆git仓库到本地

以下例子是从linux上的git服务器克隆下来

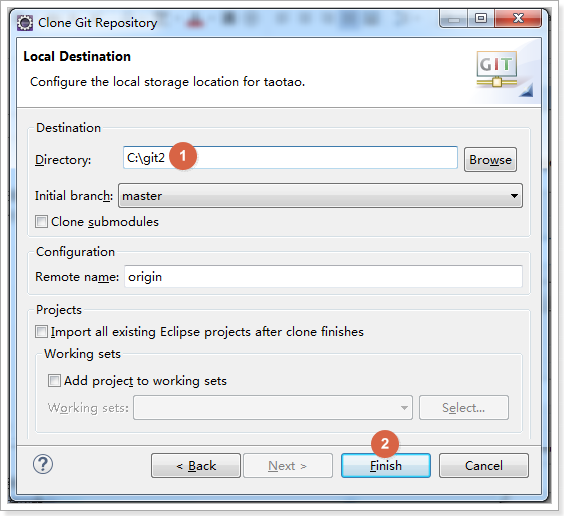






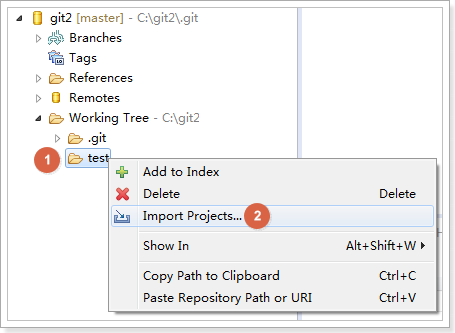
### 设置git文件存放路径

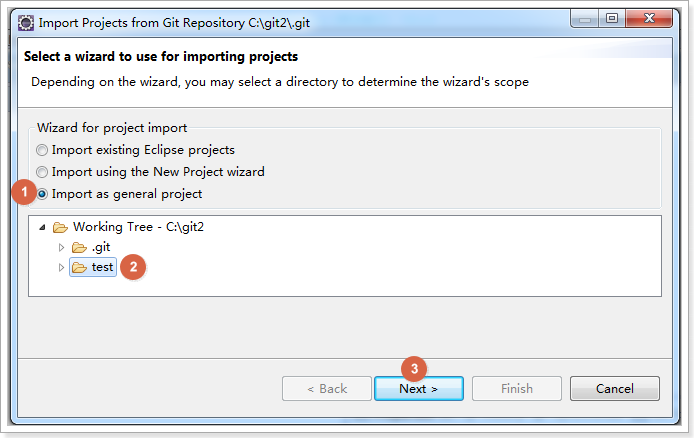
说明：这是在设置项目本地仓库的位置。



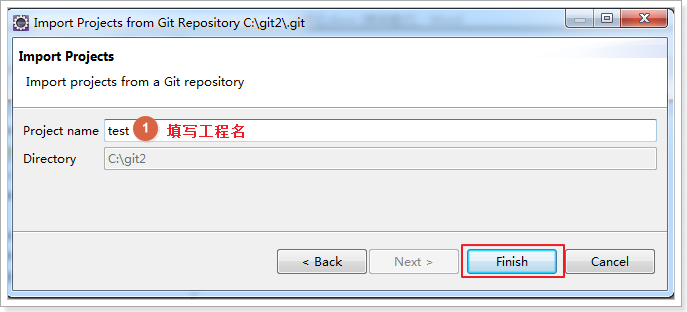
### 导出工程

说明：打开working tree目录，找到需要到处的项目右键。（一个仓库中可能有很多项目，当然这种情况很少，所以需要在指定项目中import project）

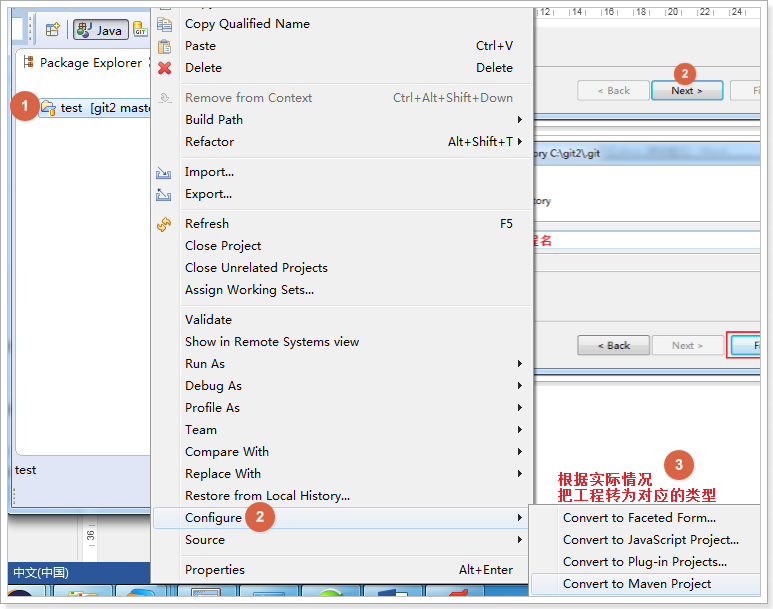




填写工程名

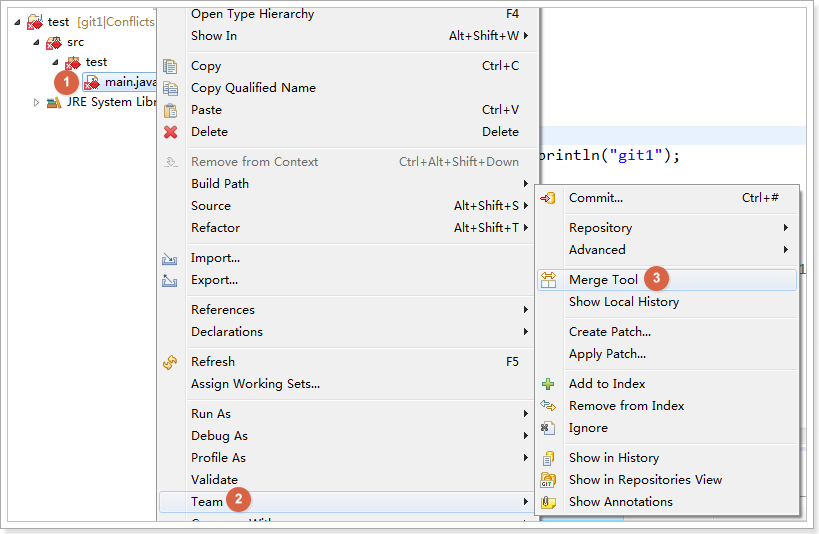


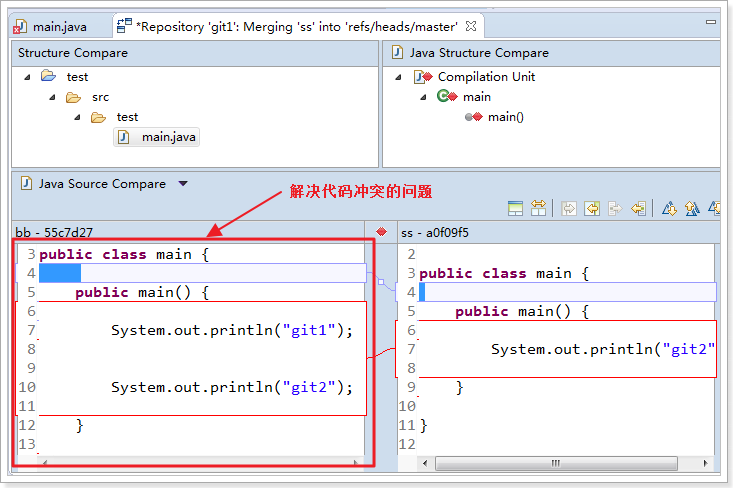
转换工程为对应的类型



## 解决冲突

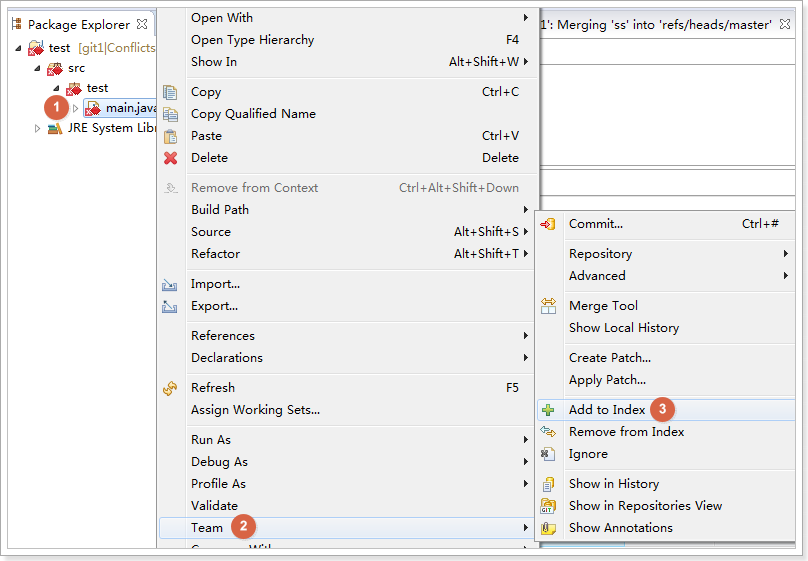
### 解决代码冲突





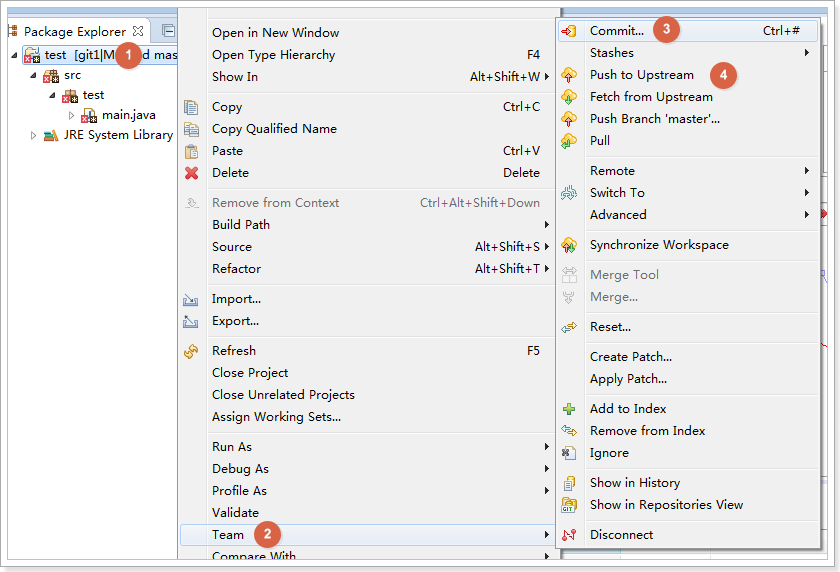
### 提交解决冲突后的代码

把解决冲突后的代码执行**Add to Index（相当于标记冲突已解决）**



先Commit到本地

再Push到远程的git服务器



# Maven工程上传下载到Github

**Github和git没有直接的联系，但是我们可以用github可以作为git的远程服务器使用。**

Git并不像 SVN 那样有个中心服务器。目前我们使用到的Git命令都是在本地执行，如果你想通过Git分享你的代码或者与其他开发人员合作。你就需要将数据放到一台其他开发人员能够连接的服务器上。

上传下载: <https://blog.csdn.net/qq_28817739/article/details/80432347>（亲测有效）

<https://blog.csdn.net/qq_28817739/article/details/80431675>

<https://blog.csdn.net/qq_28817739/article/details/80432666>

github默认不上传空目录！！！

可以先随便放一个文件在补空目录，比如.gitkeep