**一、为什么**

　　为什么不用svn？

svn是一个优秀的代码和版本管理工具，使用svn只需要搭建好svn中央仓库，配置本地svn客户端即可，自从google code关闭服务之后，互联网上已经没有非常好的公共svn仓库，如果是公司内部使用，还可以用上公司内网搭建的svn服务器，但个人就没那么方便了。

　　为什么用git？

git和svn同样作为代码和版本管理工具，git的理念更先进，而且公共git仓库[github](https://github.com/" \o "https://github.com/" \t "_blank)真的好用，众多优秀的开源项目都放在github

　　什么是git、github、tortoisegit？

github是目前世界上最大的代码托管平台，对个人也提供了非常丰富的功能；[git客户端](https://www.git-scm.com/" \o "https://www.git-scm.com/" \t "_blank)是windows系统下与github仓库传输数据的客户端工具；[tortoisegit](https://tortoisegit.org/" \o "https://tortoisegit.org/" \t "_blank)是git图形管理界面，依赖git客户端，提交时可以清晰看到变更内容

**二、怎么用**

**1、注册github账号**

　　github官方网站：[https://github.com/](https://github.com/" \o "https://github.com/" \t "_blank)，注册过程：略

**2、安装git客户端与tortoisegit**

　　git客户端下载地址：[https://www.git-scm.com/](https://www.git-scm.com/" \o "https://www.git-scm.com/" \t "_blank)

　　tortoisegit下载地址：[https://tortoisegit.org/](https://tortoisegit.org/" \o "https://tortoisegit.org/" \t "_blank)

　　双击下载的安装包，默认安装直到完成。

**3、配置git客户端**

　　git客户端需要配置ssh建立与github之间的远程安全连接，安装完git客户端之后有如下快捷方式

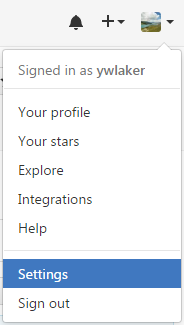


　　打开git bash，输入以下内容，邮箱为github注册邮箱

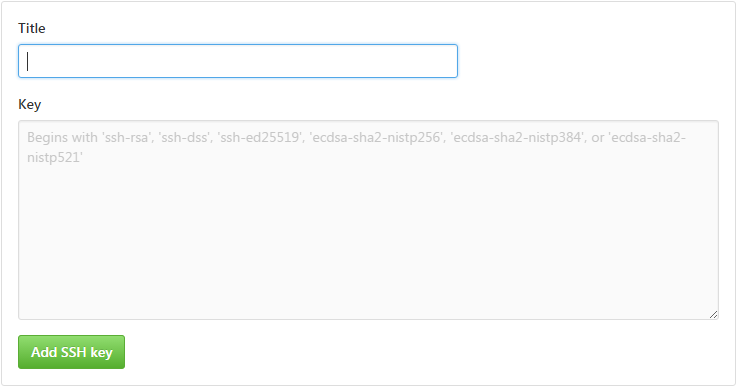
|  |  |
| --- | --- |
| 1 | ssh-keygen -t rsa -C "573215750@qq.com" |

　　回车，输入“yes”，再回车，将在“C:\Users\john\.ssh\id\_rsa.pub”文件生成ssh密钥，复制所有内容

　　登录github，打开“Settings”



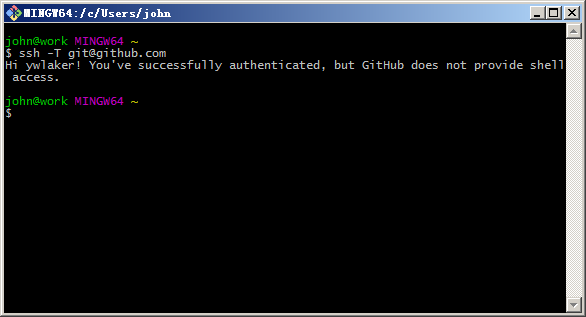
　　点击左侧“SSH and GPG keys”菜单，然后点击右侧上方“New SSH key”按钮，“Title”随意输入名称，“Key”处粘贴刚才复制的密钥，注意最后不要留空格或换行，保存配置



　　打开git bash，输入以下内容验证ssh连接

|  |  |
| --- | --- |
| 1 | ssh -T git@github.com |

　　第一次需要输入“yes”然后回车才能看到连接成功的信息，以后就不需要了



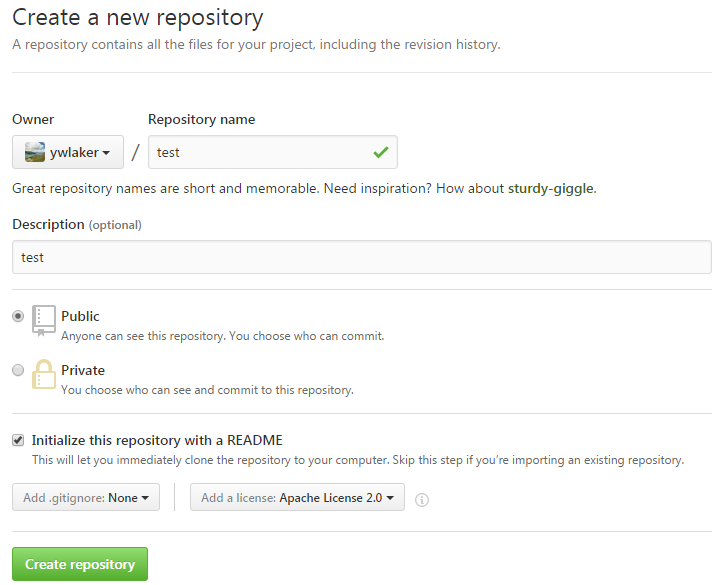
**4、配置全局信息**

　　配置全局的用户名、邮箱信息，作为提交人信息

|  |  |
| --- | --- |
| 1  2 | git config --global user.name "ywlaker"  git config --global user.email "573215750@qq.com" |

**5、创建github仓库**

　　登录github，创建一个工程，名为“test”，勾选“Initialize this repository with a README”，并选择一个license，以“Apache License 2.0”为例



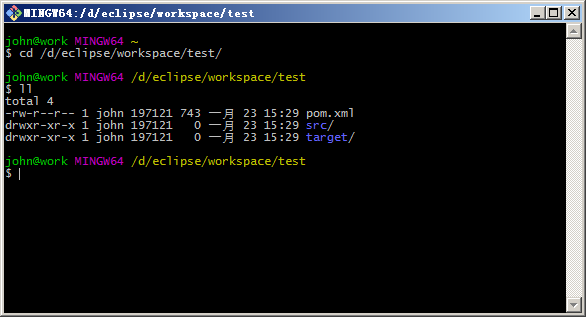
　　创建完“test”仓库之后，查看该仓库，只包含两个文件：“README.md”和“LICENSE”

**6、创建eclipse工程**

　　在eclipse下创建工程，名为“test”（名称任意），在资源管理器中查看项目



　　在git bash中查看项目（linux下点开头的文件为隐藏文件，所以只能看到三个文件（夹））



**7、同步eclipse工程与github仓库**

　　github的license与readme文件要同步到eclipse工程，eclipse工程的代码要同步到github仓库，并且eclipse自动生成的那些“.settings”、“.project”、“.classpath”还有“target”不需要同步到github仓库，怎么处理？

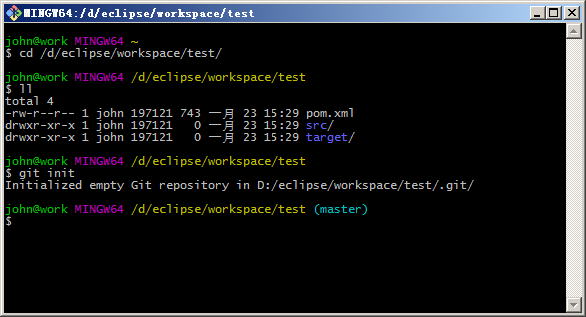
**7.1、git初始化eclipse工程目录**

　　在资源管理器中打开“test”工程根目录，右键“Git Create repository here...”



　　或者，git bash输入命令

|  |  |
| --- | --- |
| 1  2 | cd /d/eclipse/workspace/test/  git init |



**7.2、同步github仓库文件到eclipse工程目录**

　　git bash输入命令

|  |  |
| --- | --- |
| 1  2 | git remote add origin git@github.com:ywlaker/test.git  git pull origin master |

**7.3、添加.gitignore**

　　将eclipse自动生成的与代码无关的文件排除在同步列表之外，git bash切换到工程根目录，输入

|  |  |
| --- | --- |
| 1 | vi .gitignore |

　　然后按下键盘的“i”键，输入

|  |  |
| --- | --- |
| 1  2  3  4 | target/  .settings  .classpath  .project |

　　按下“esc”键，再输入“:wq”保存退出，此时“test”工程根目录多了一个“.gitignore”文件

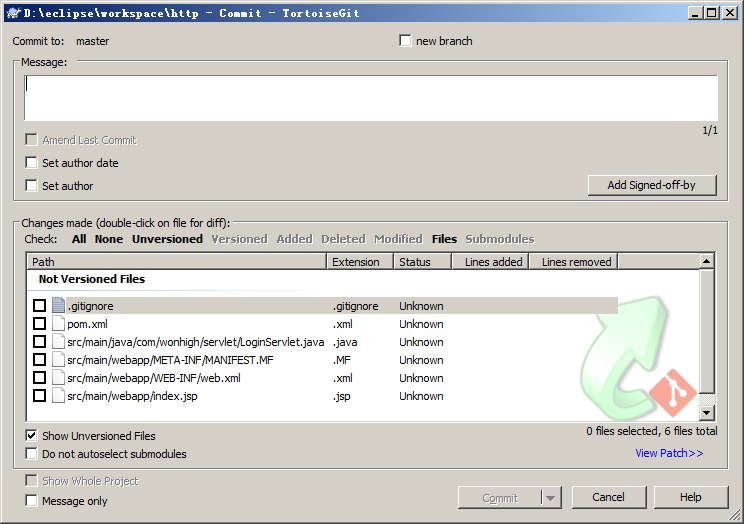
　　因为windows下无法新建以点号开头的文件，所以只能用上面这种方式。

**7.4、同步eclipse工程到github仓库**

　　在资源管理器中进入“test”工程根目录，右键“Git Commit -> 'master'”



　　出现和tortoisesvn一样的提交界面，确定内容后提交即可



　　这一步是将文件提交到本地git缓存，用命令行也可以，git bash切换至工程更目录，输入

|  |  |
| --- | --- |
| 1  2 | git add ./\*  git commit -m "comment" |

　　最后，将本地缓存同步到github仓库

|  |  |
| --- | --- |
| 1 | git push -u origin master |

**Push失败：**

今天想把最近的一些小代码push到github上

我先是将该目录git init进行初始化，然后

git add -A ，

接着

git commit -m “description...”

然后在github上新建一个远程库，想要将其与我本地的库相关联，使用：

git remote add origin git@github.com:YOURGITHUBNAME/REPOSITORY.git

并push到远程端

至此，一切顺利

于是，我使用了命令：

git push -u origin master

结果报错如下：

error: failed to push some refs to 'git@github.com:你的远程库名.git'  
hint: Updates were rejected because the tip of your current branch is behind  
hint: its remote counterpart. Integrate the remote changes (e.g.  
hint: 'git pull ...') before pushing again.  
hint: See the 'Note about fast-forwards' in 'git push --help' for details.

从提示可以看出，是由于两者不同步，因此需要先pull，进行合并然后在进行push，

因此先使用

git pull --rebase origin master

将远程文件同步下来。

然后再执行推送

git push -u origin master

终于，成功！

添加错误的远程舱，删除远程舱：

git remote rm origin