目录

[背景](#_Toc24076_WPSOffice_Level1) [3](#_Toc24076_WPSOffice_Level1)

[1. selenium+java编码及调试](#_Toc27182_WPSOffice_Level1) [3](#_Toc27182_WPSOffice_Level1)

[2. 代码提交Github进行管理](#_Toc21167_WPSOffice_Level1) [14](#_Toc21167_WPSOffice_Level1)

[3. 借用jenkins实现自动构建](#_Toc27948_WPSOffice_Level1) [24](#_Toc27948_WPSOffice_Level1)

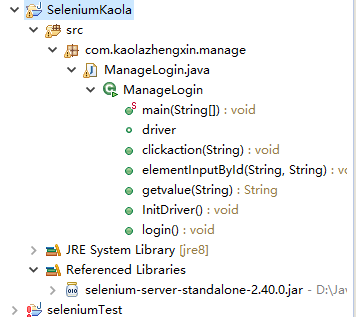
# 背景

实践selenium+java自动化编码，并使用github进行代码管理，最终用jenkin实现自动构建。通过以上内容熟悉自动化整个过程。

# selenium+java编码及调试

## 1.1代码说明

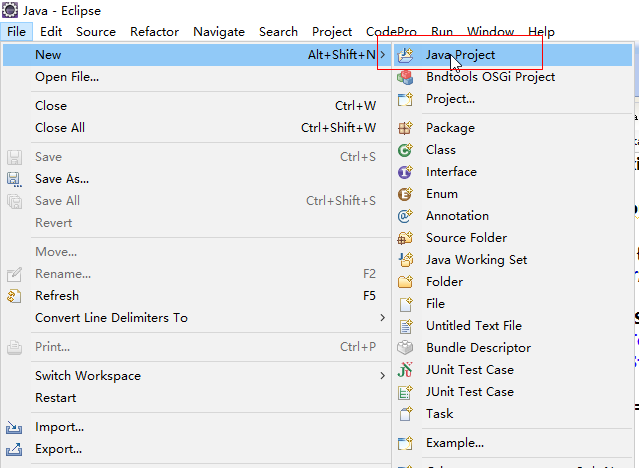
本次代码导入eclipse查看如下图：

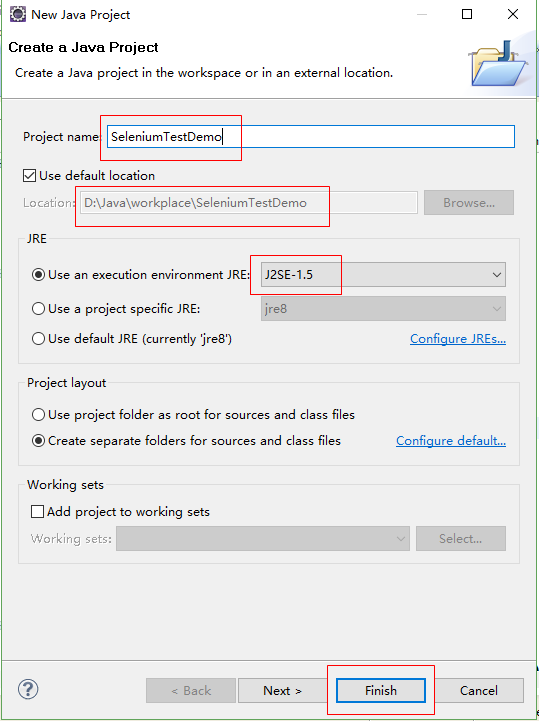


代码内容见：\jenkins\SeleniumKaola

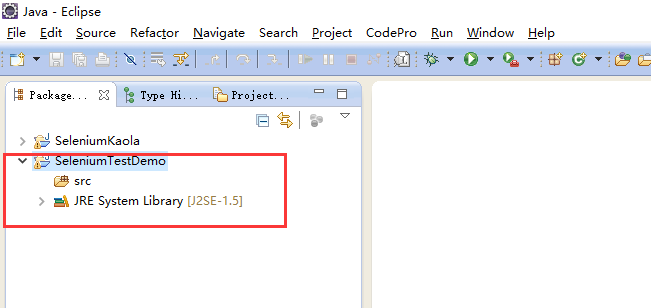
## **1.2 eclipse及调试过程**

1. 打开eclipse，新建一个工程

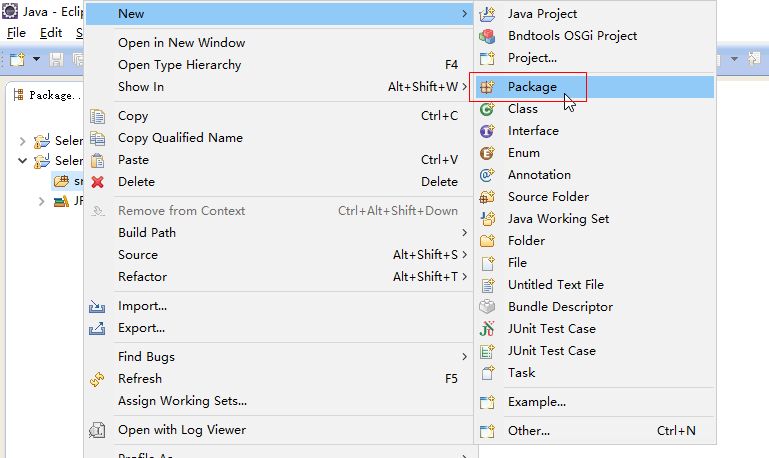
File-new-java project

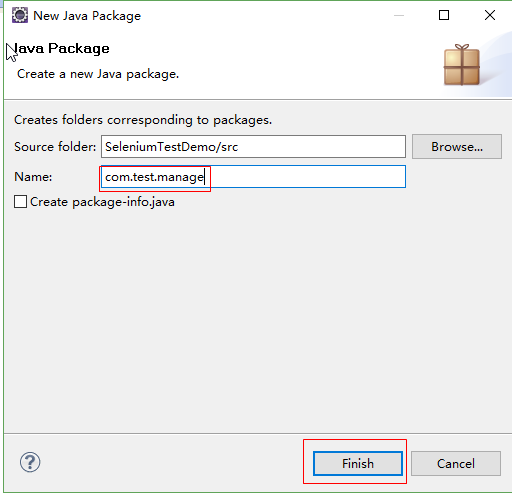


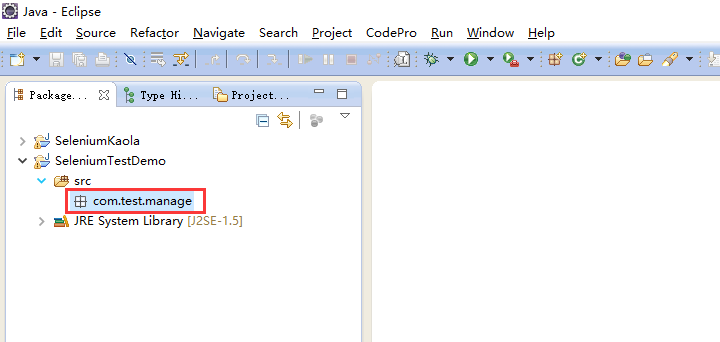
结果如下：



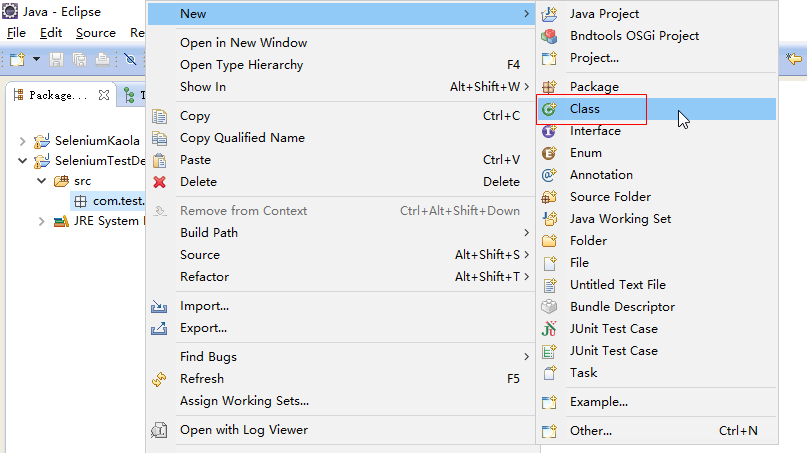
1. 在src下新建一个package

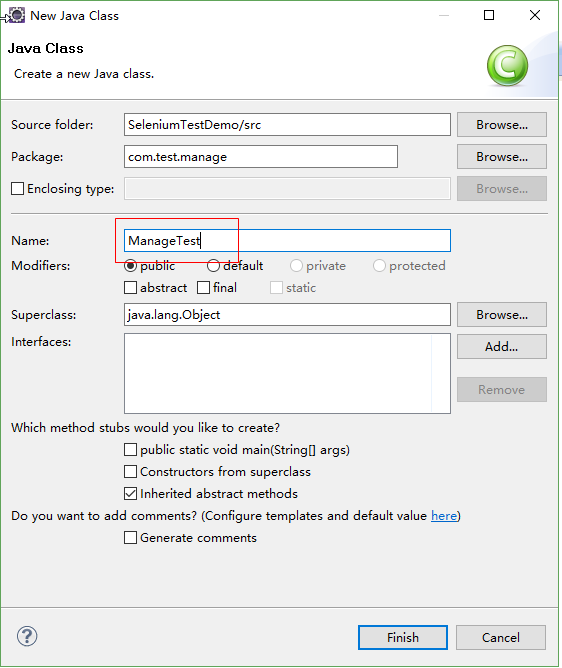


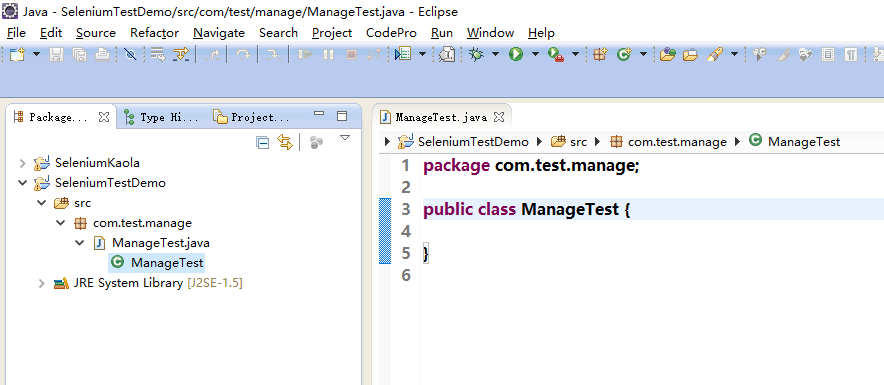




1. 在com.test.manage下新建一个class

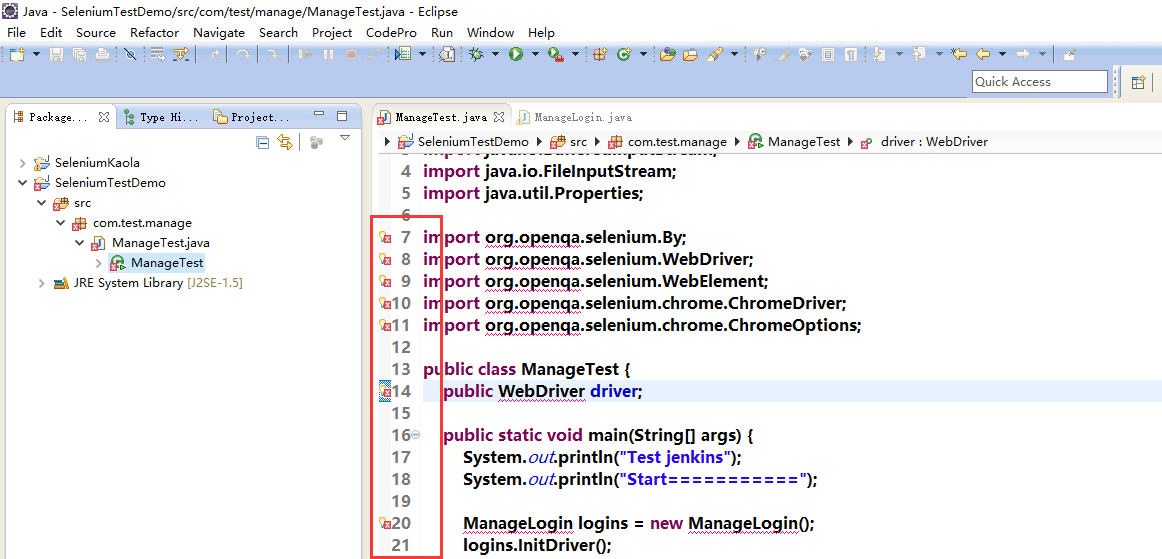






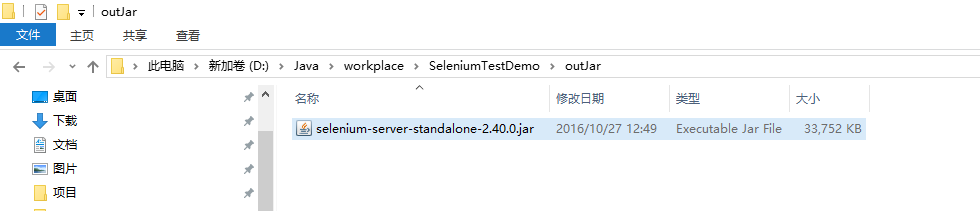
1. 在class里编写打开浏览器的代码

我这里是将之前写好的代码复制过来的，因此导包报错

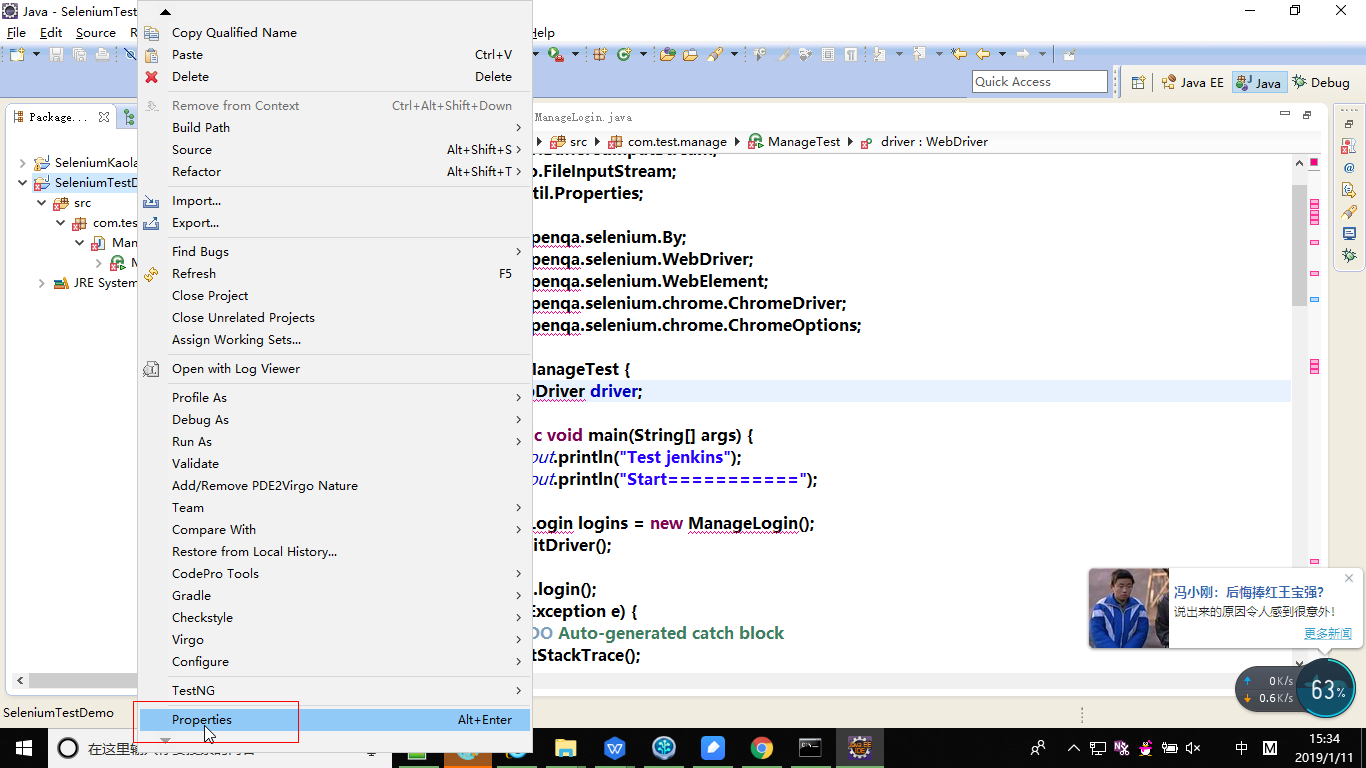


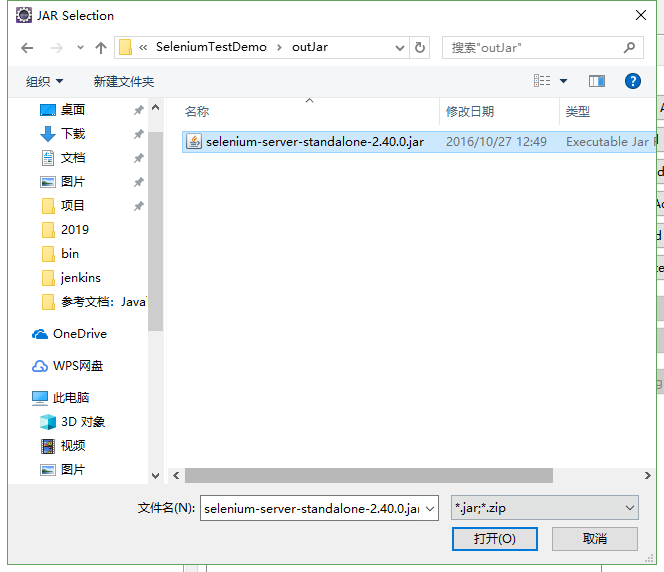
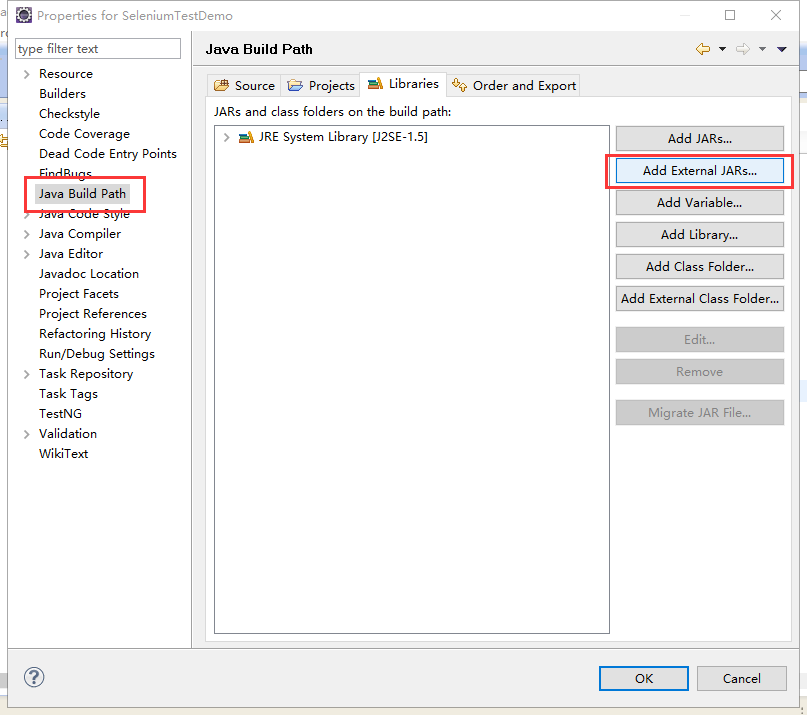
处理方法：引入selenium相关的jar包，这里用的是selenium-server-standalone-2.40.0.jar

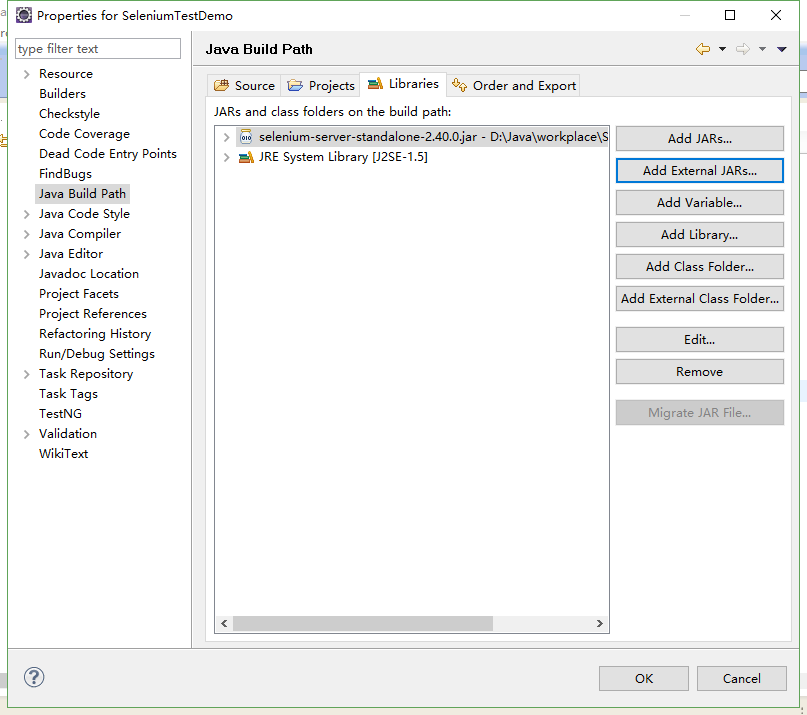
为方便后续jenkins任务执行，将jar包放到与工程一致的路径下，在D:\Java\workplace\SeleniumTestDemo下新建一个文件夹取名为：outJar，并放入事先下载好的jar包



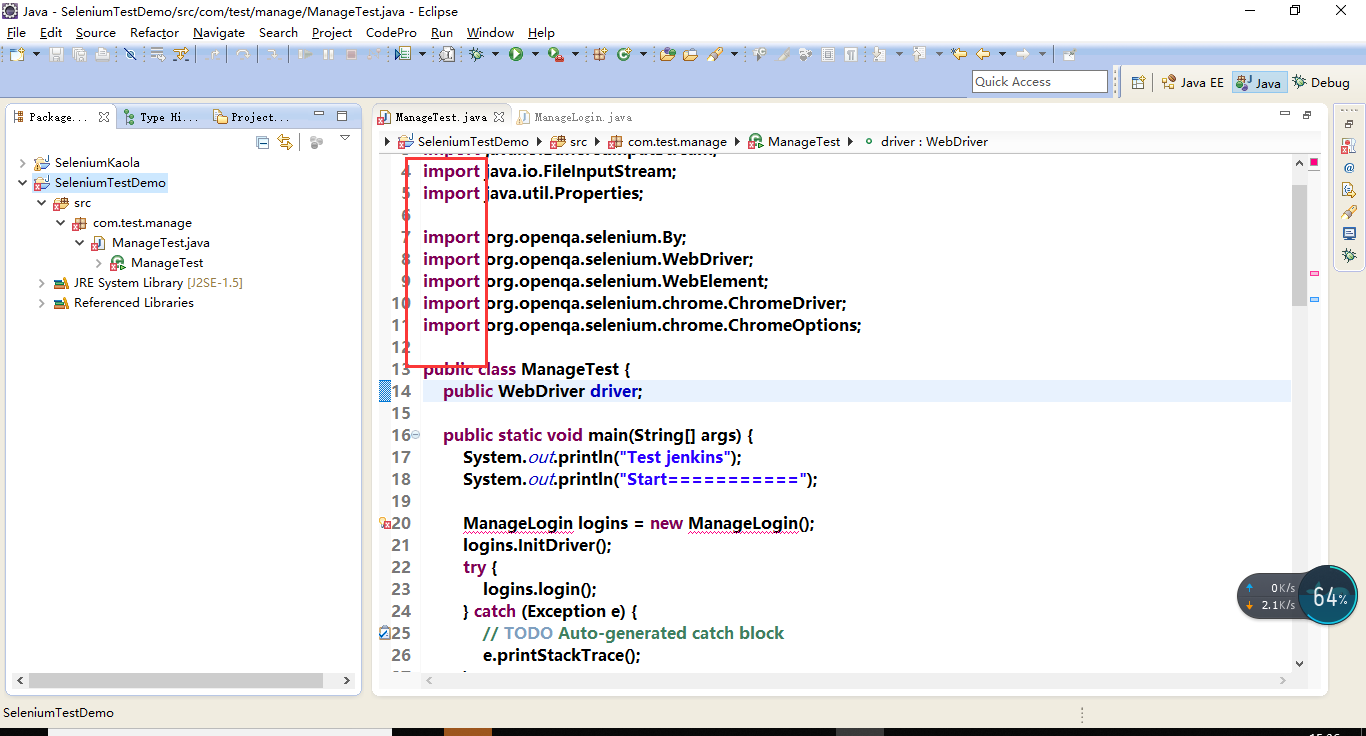
接下来在eclipse中进行配置



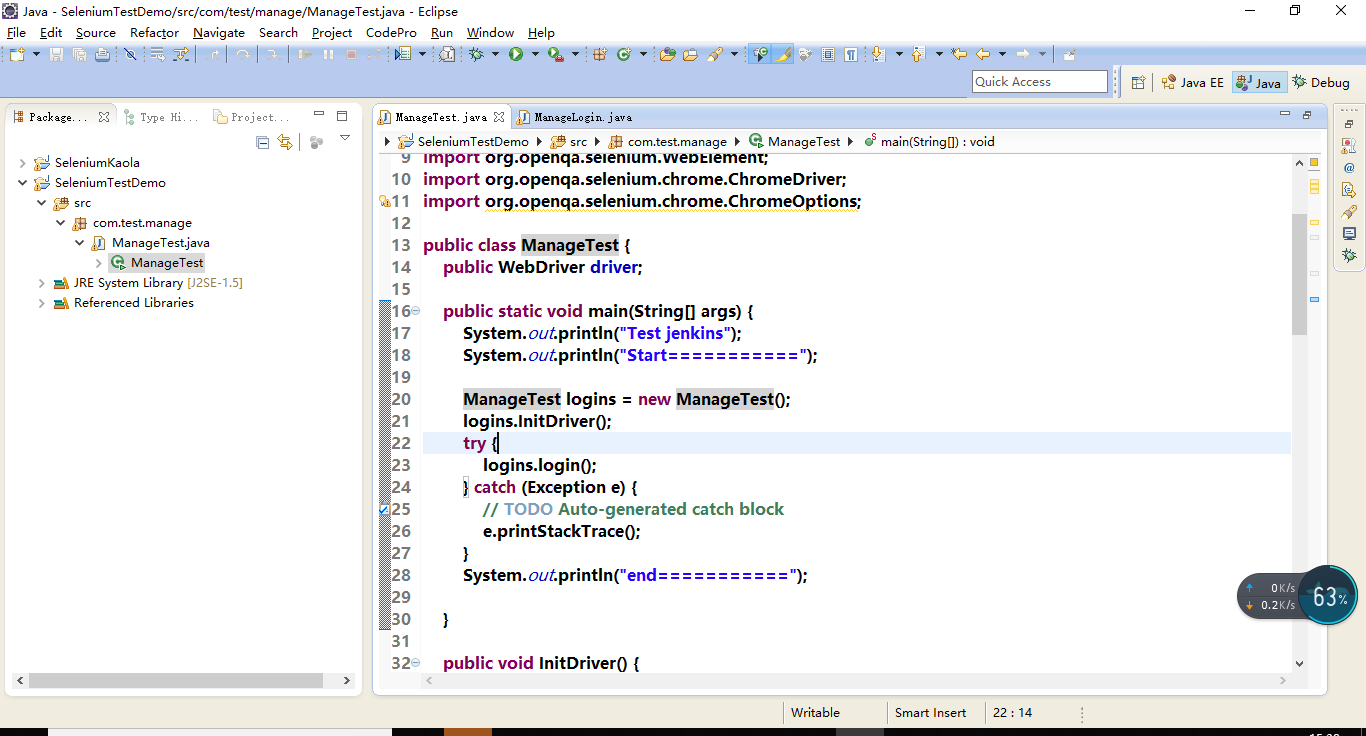


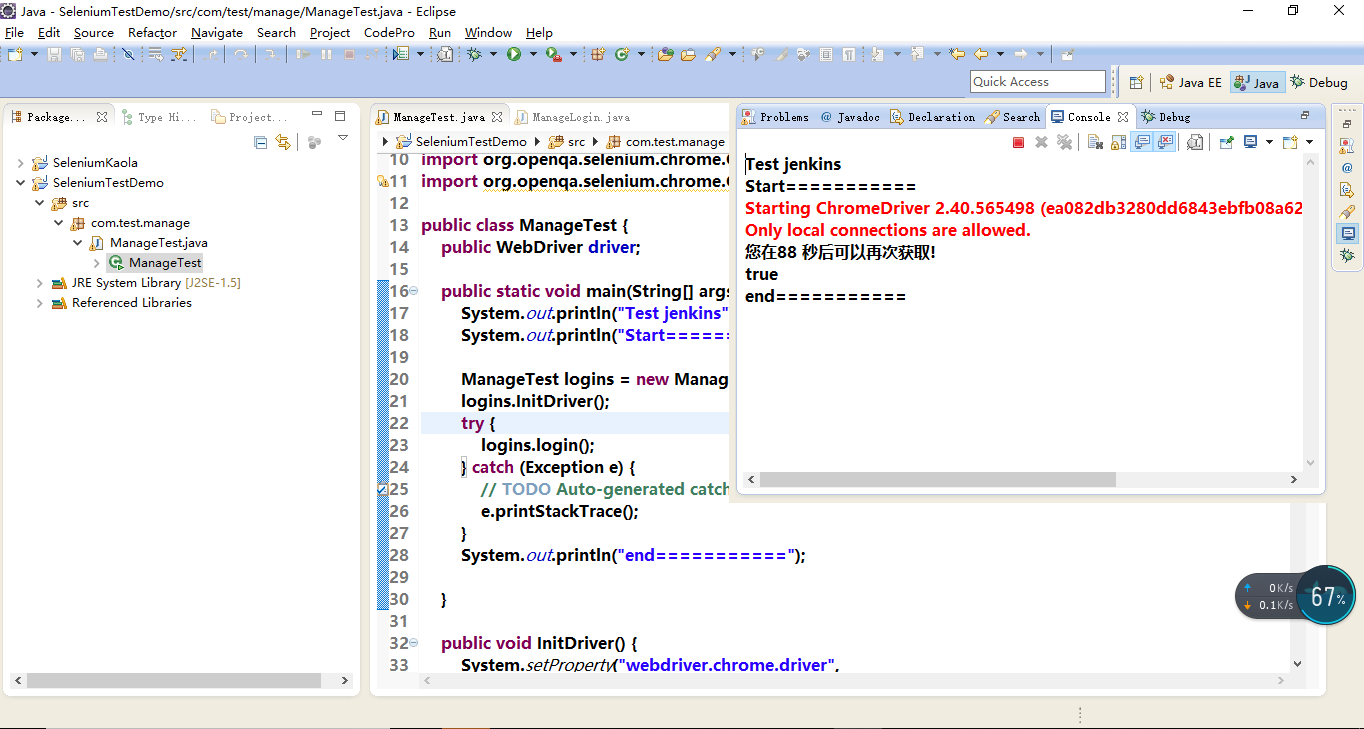


之前的导包报错就消失了



1. 因为是之前的代码，class名等不一致，稍微进行修改并执行



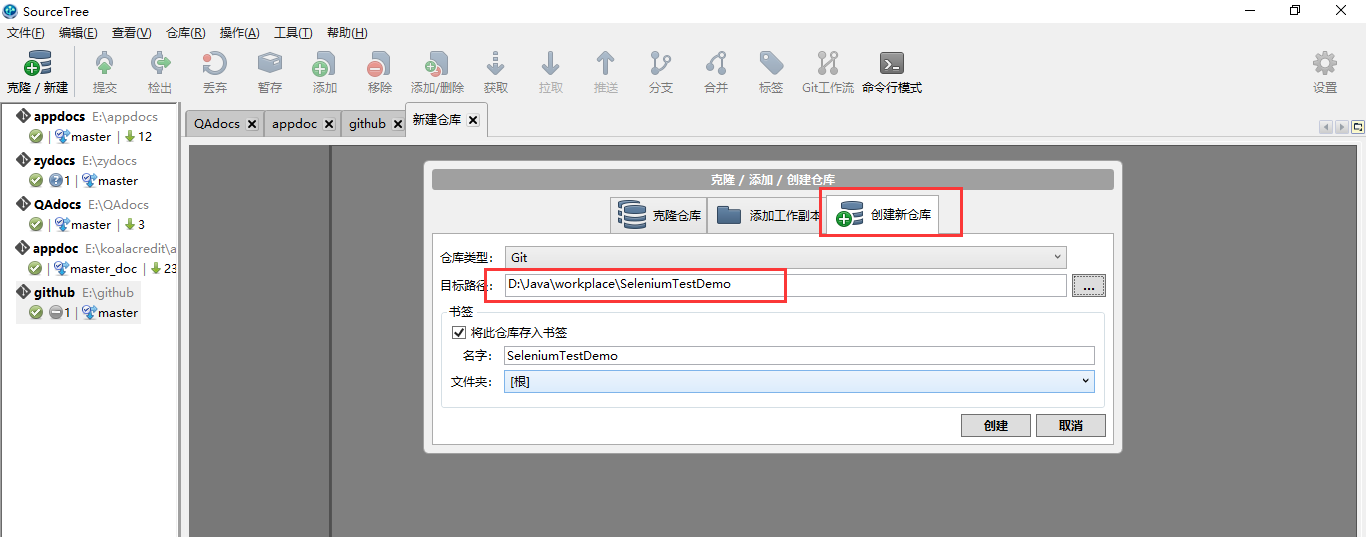


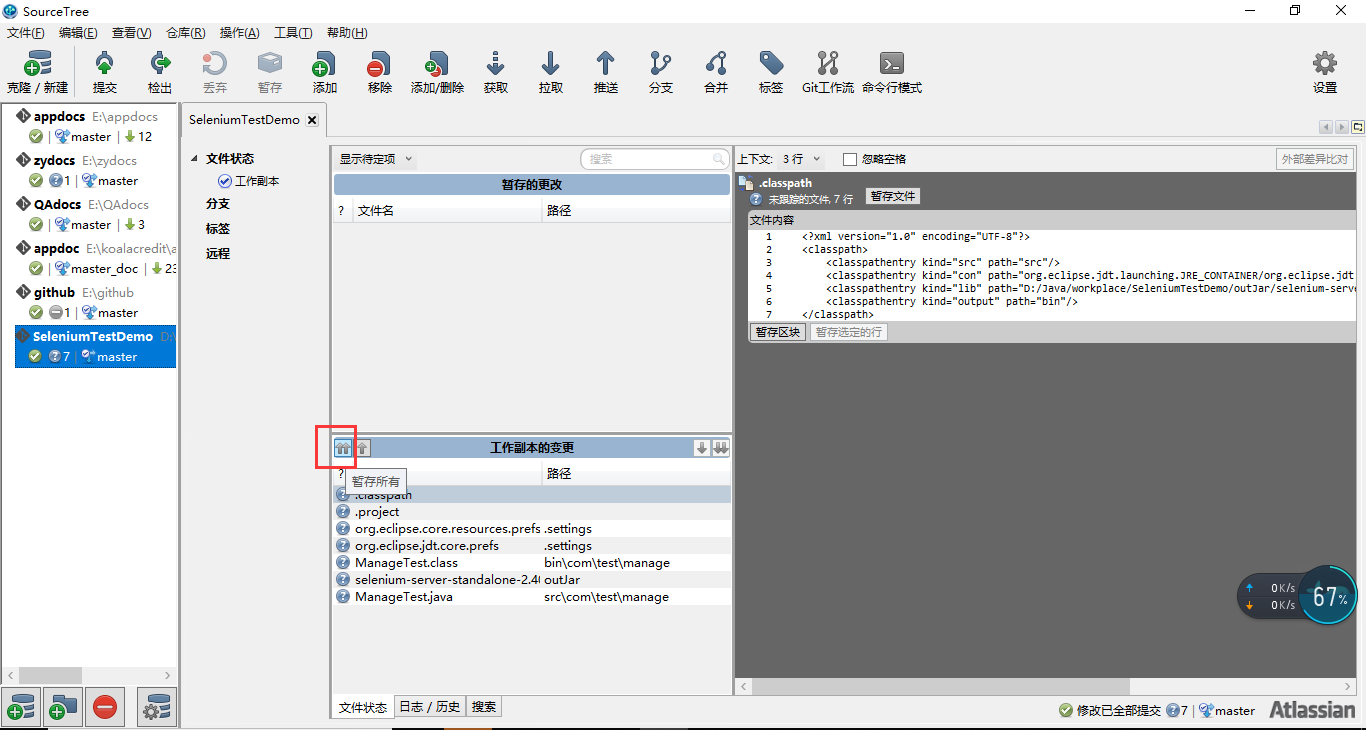
至此，代码调试成功。

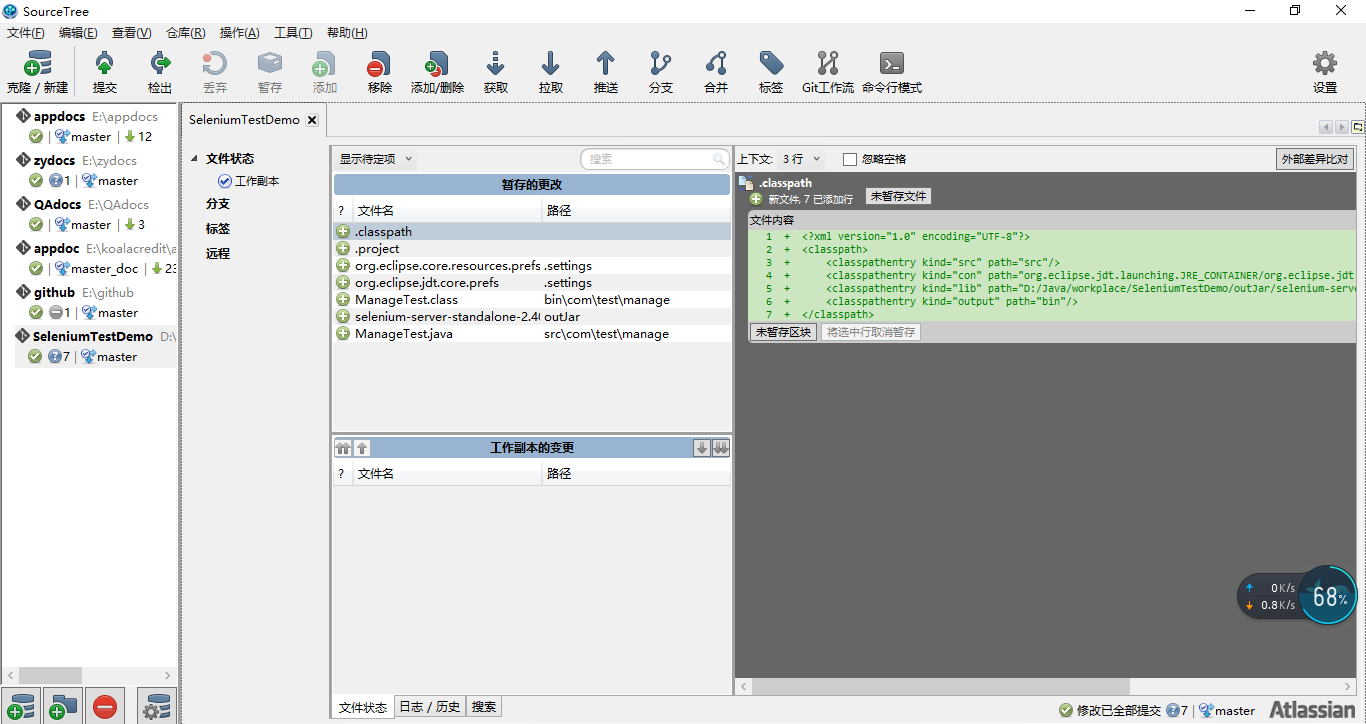
# 代码提交Github进行管理

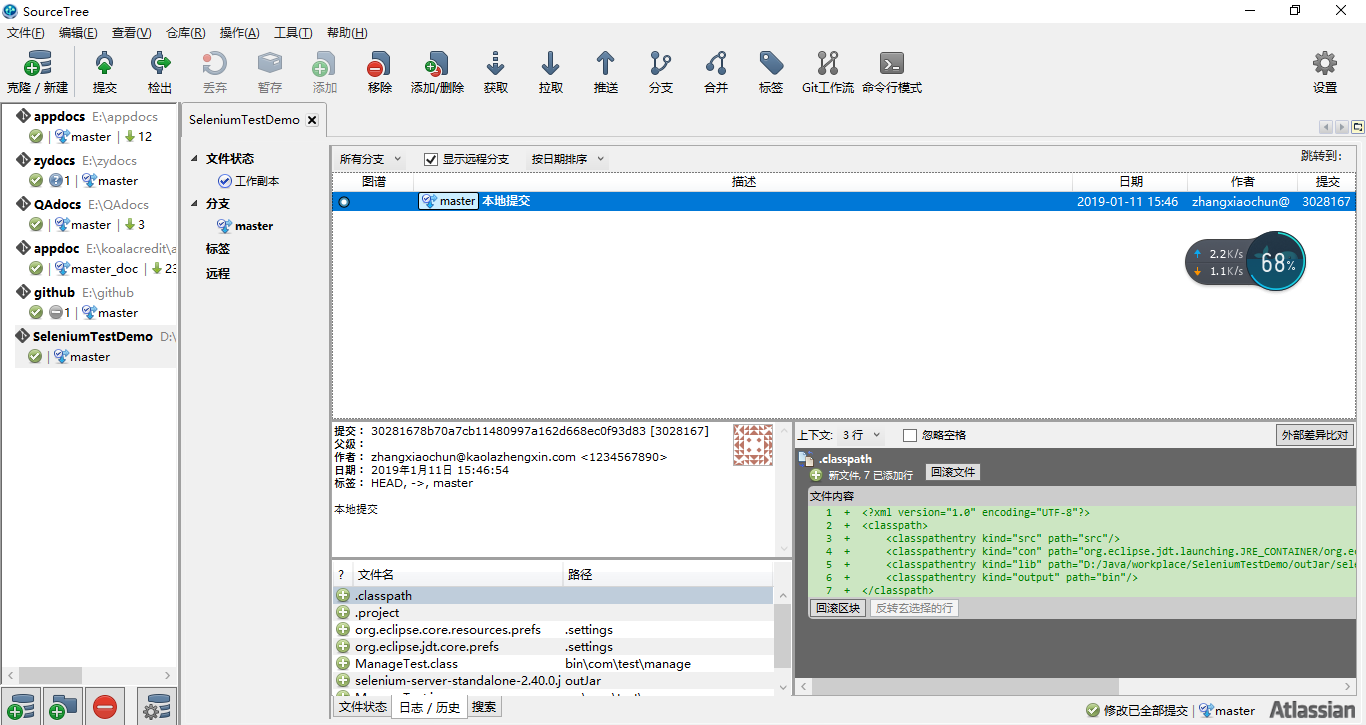
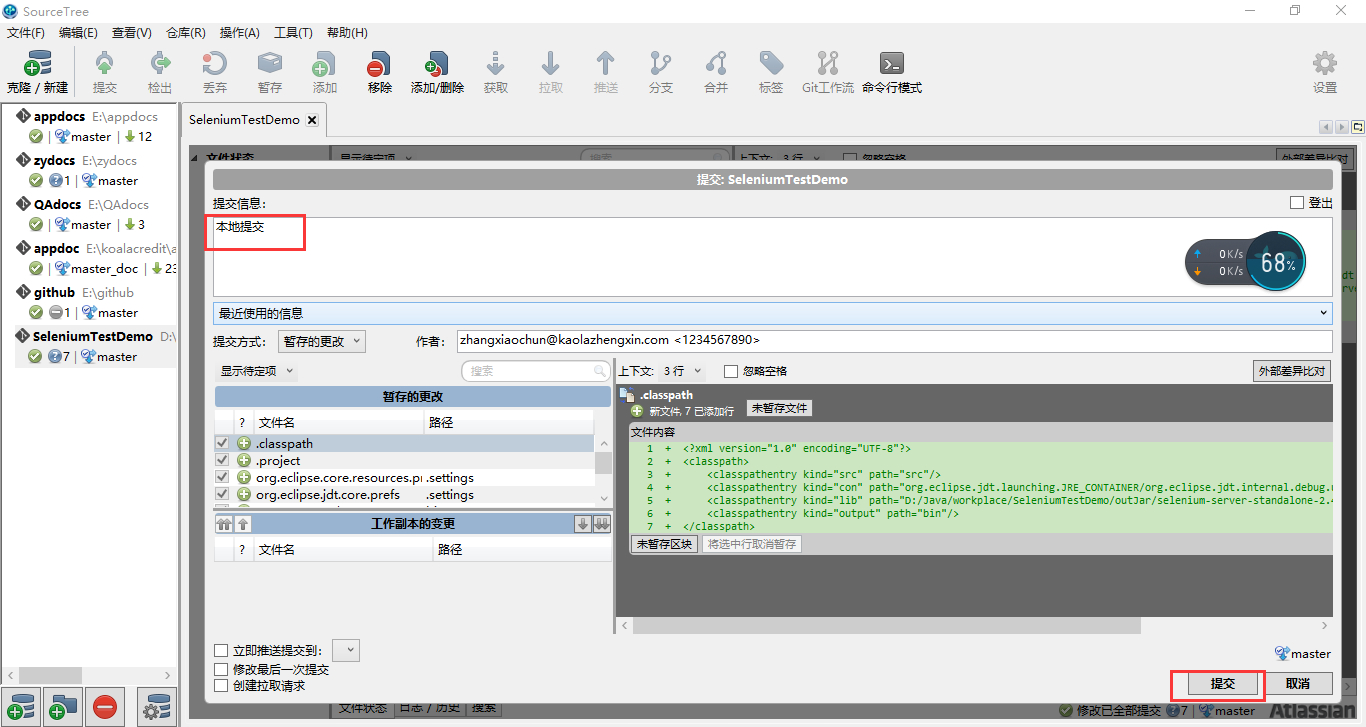
创建本地仓库并跟github进行关联，过程可以使用命令行也可以使用sourceTree

## **2.1新建本地仓库**



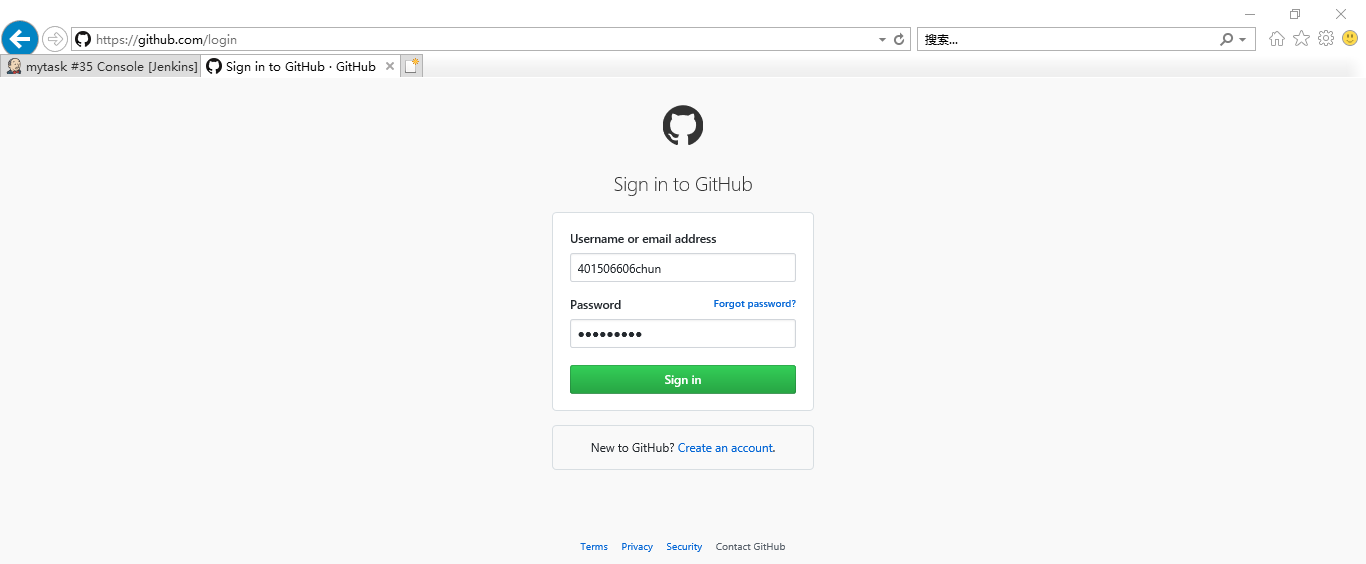


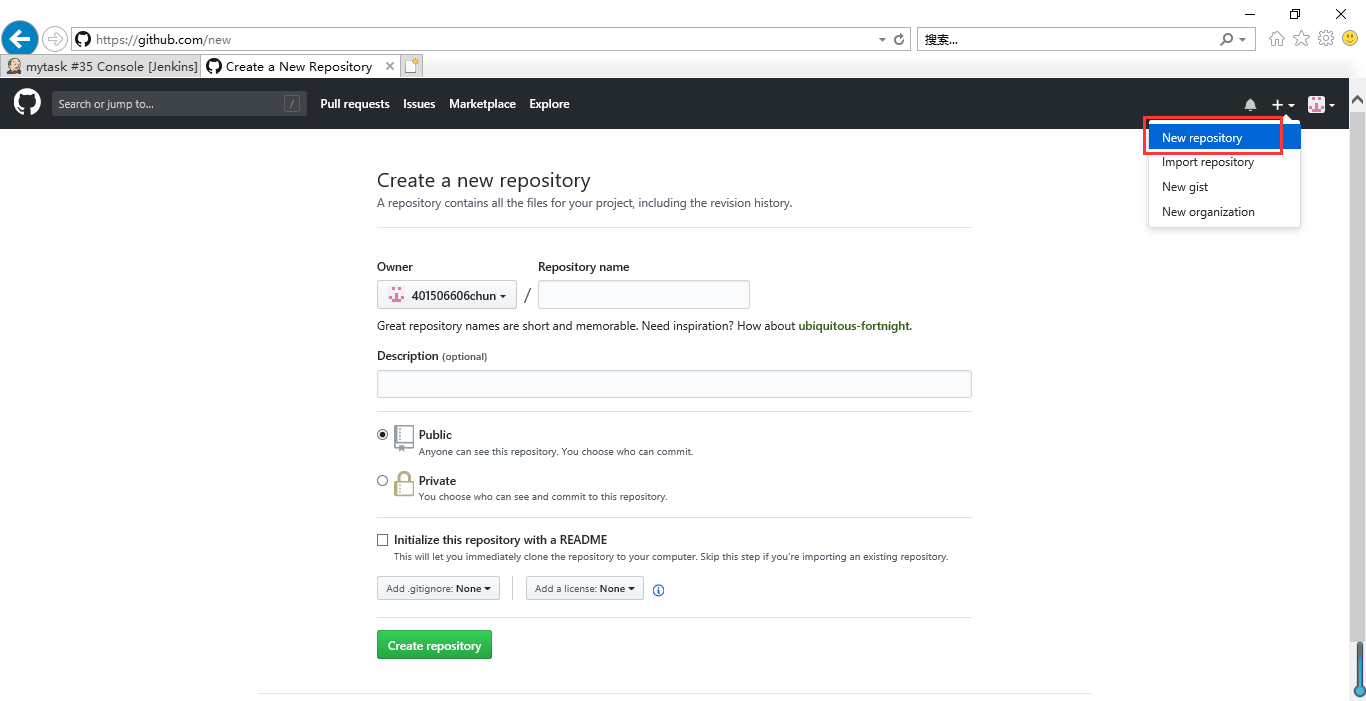


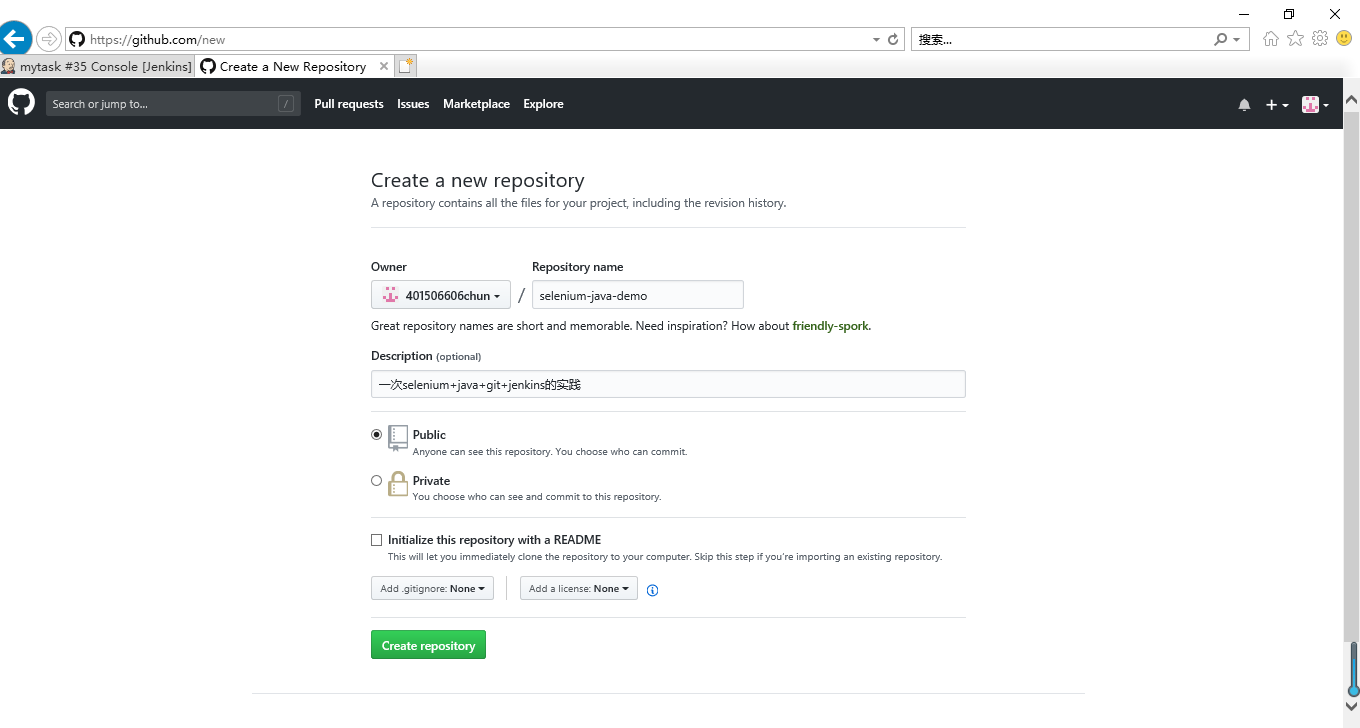


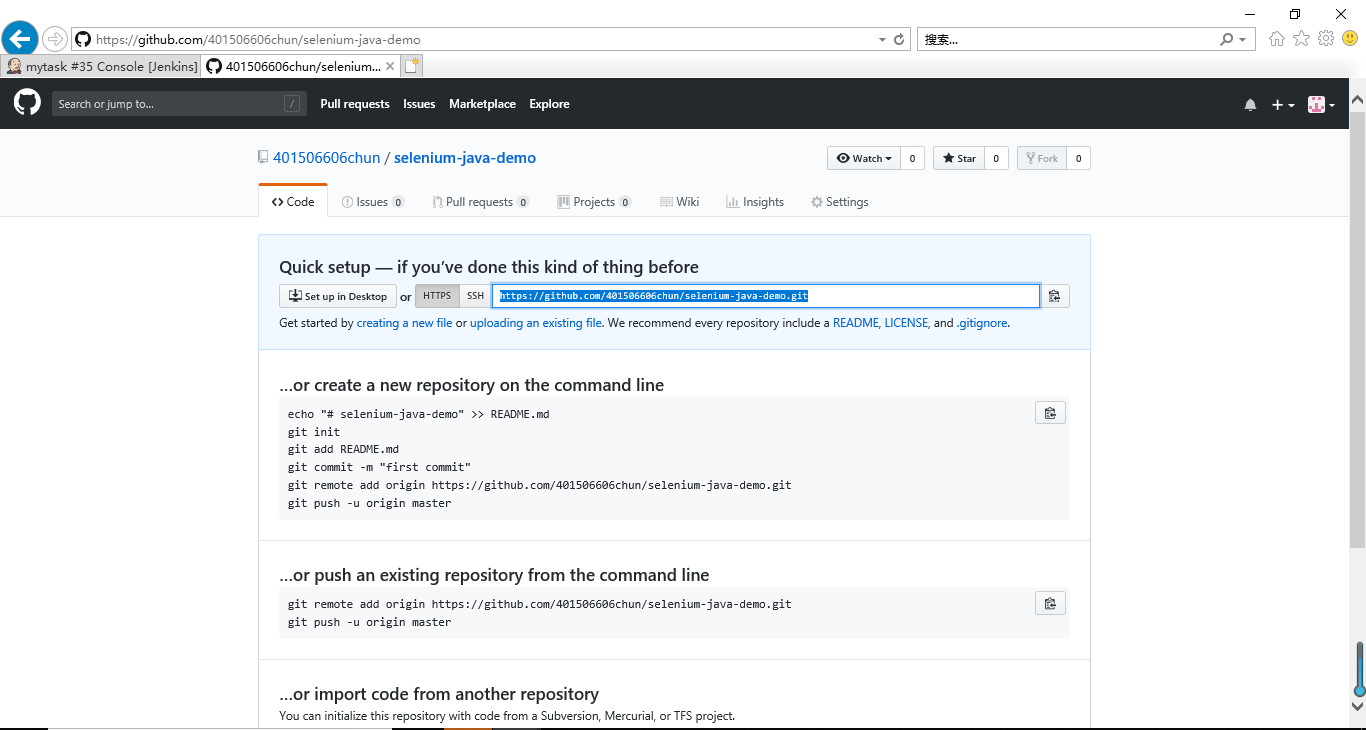
## **2.2本地与github远程仓库关联**

### 2.2.1Github登录成功并创建



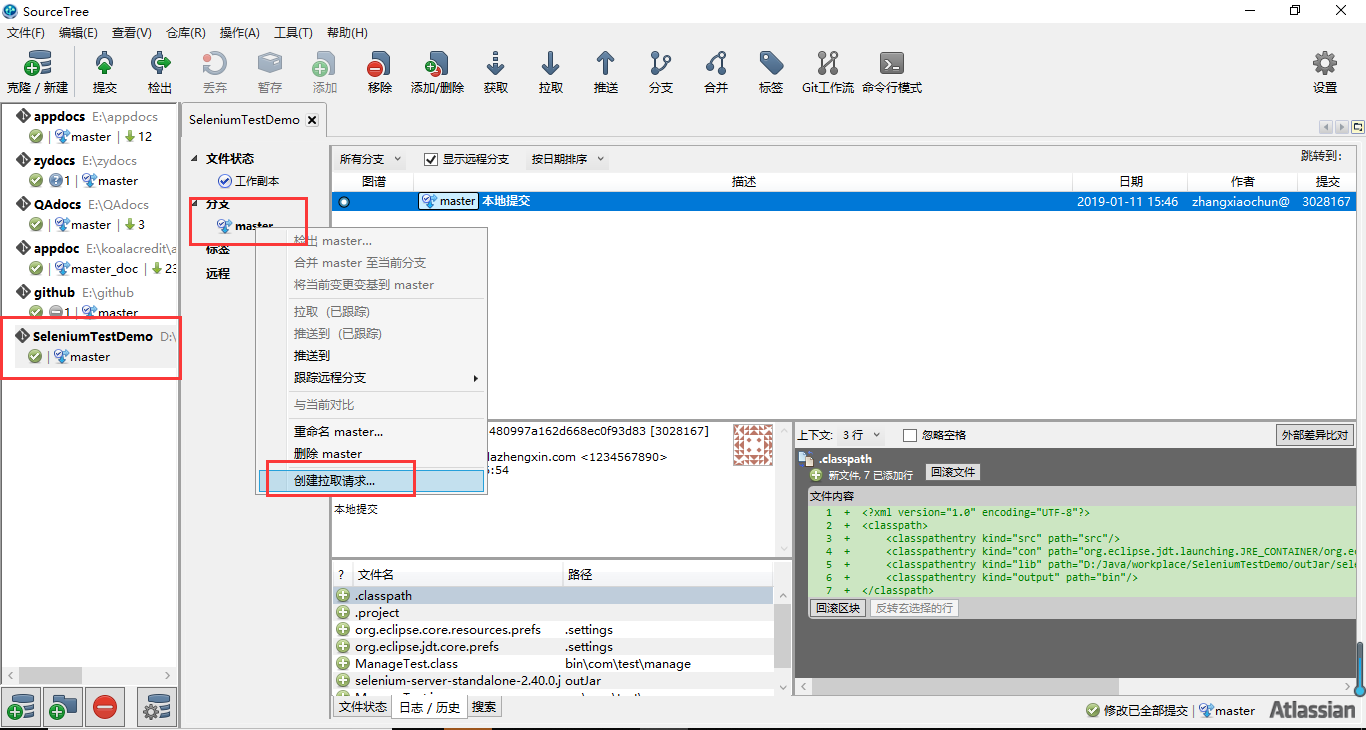


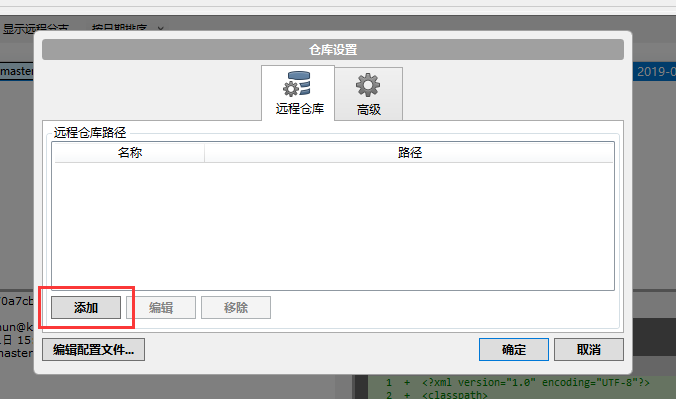


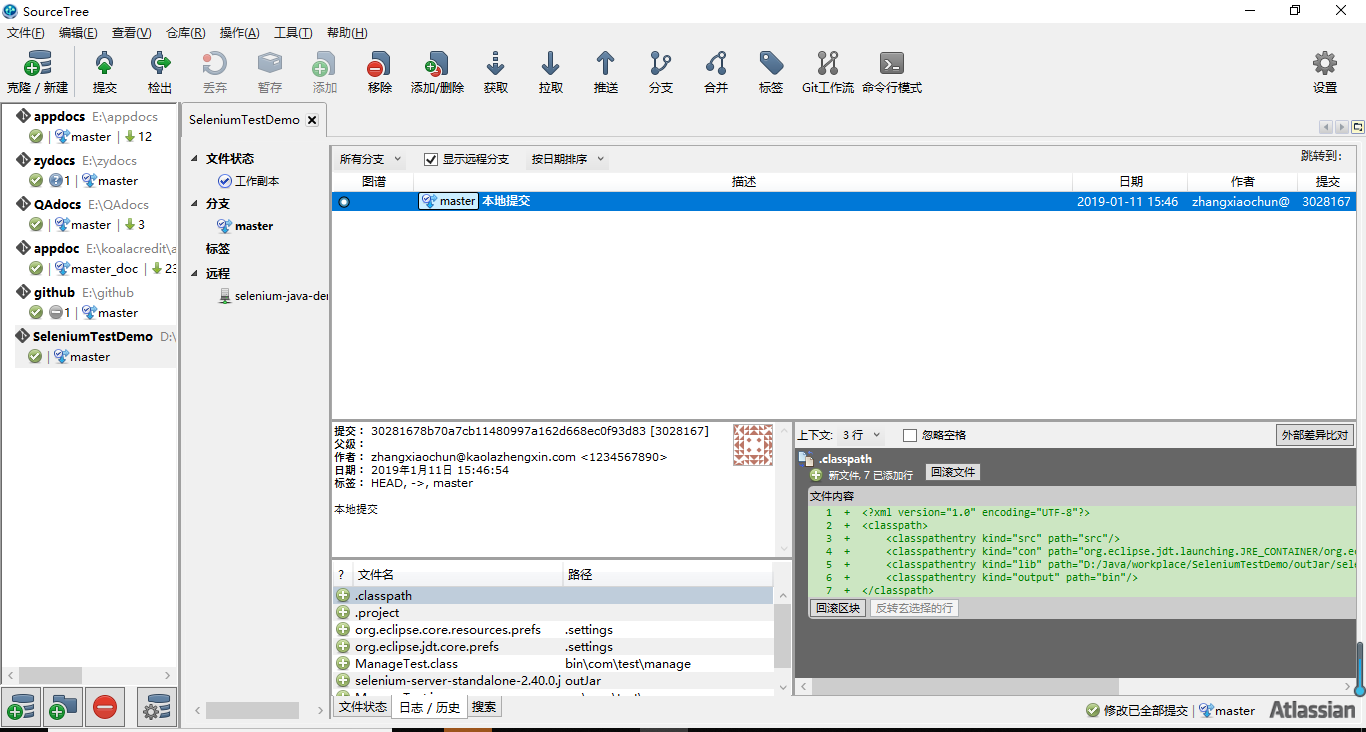
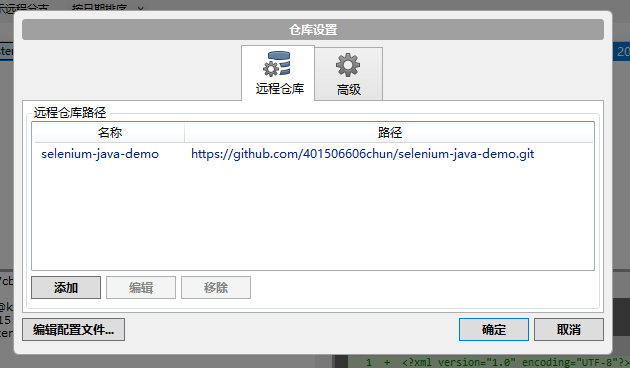
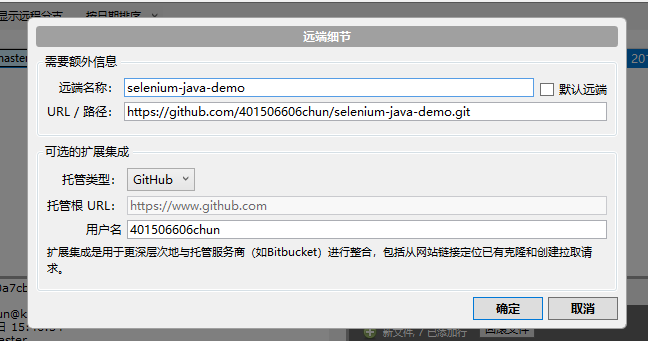


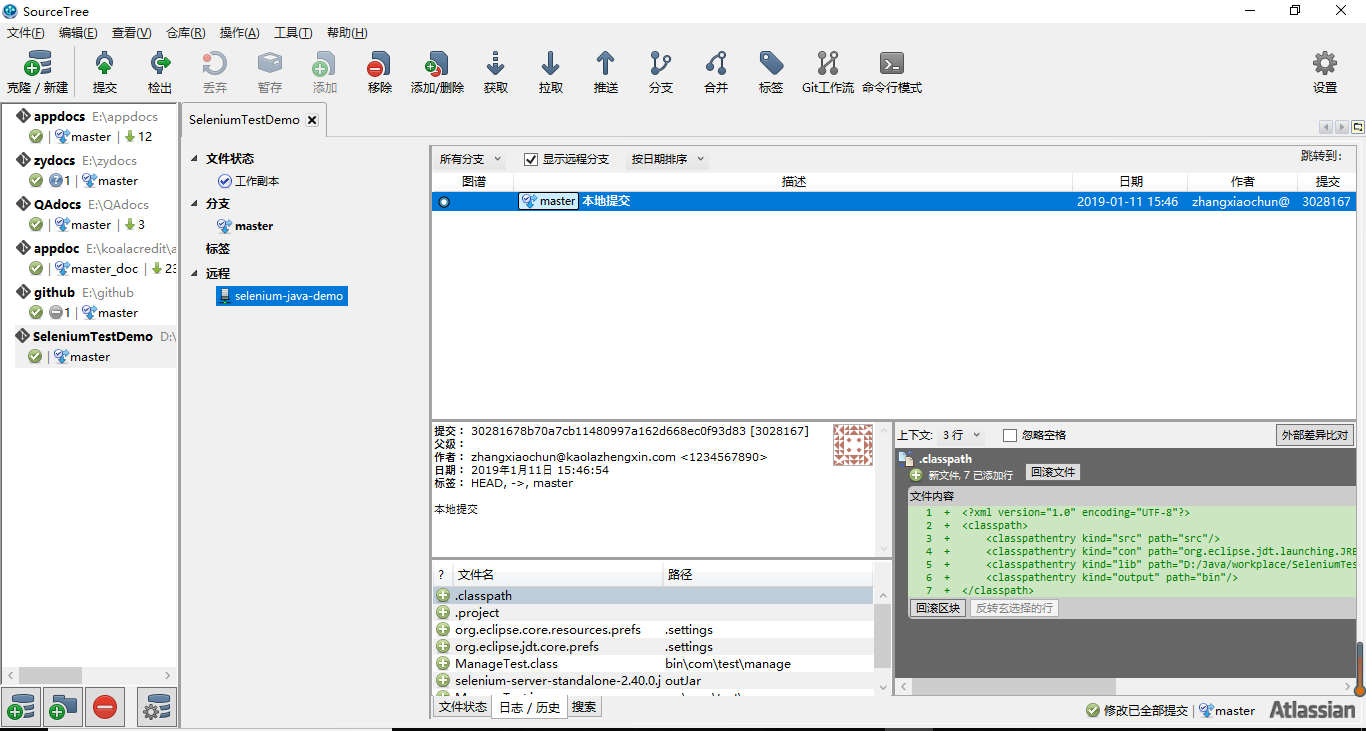
复制地址，建立本地与远程的关联

### 2.2.1**建立本地与远程的关联**

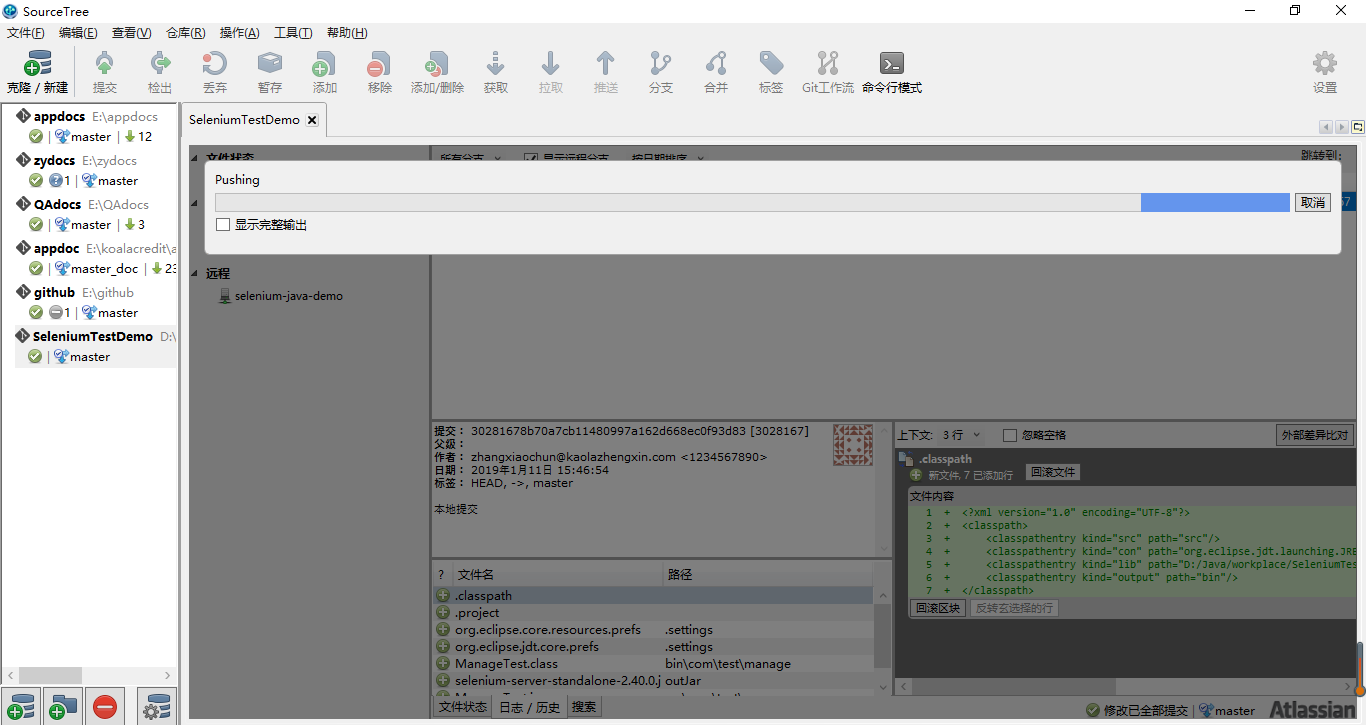
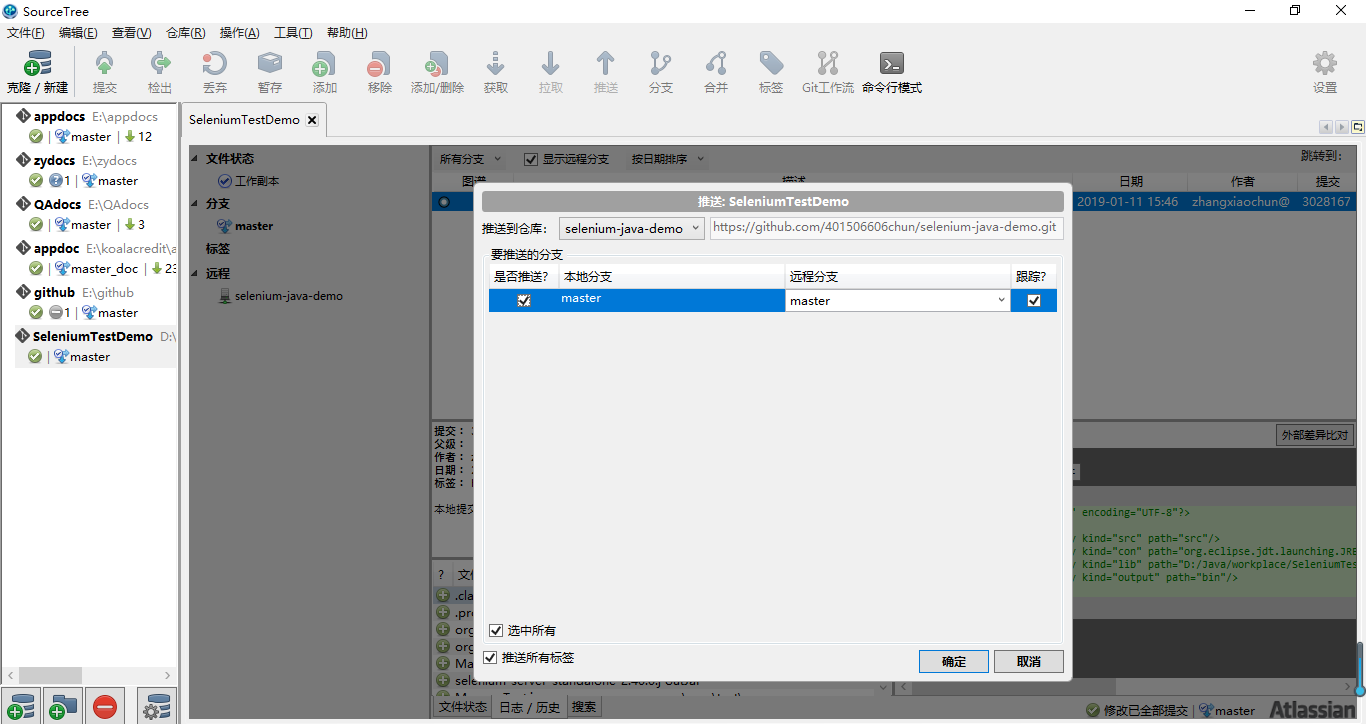
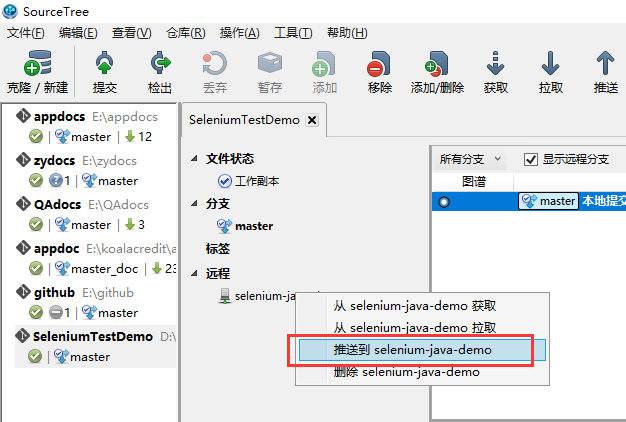


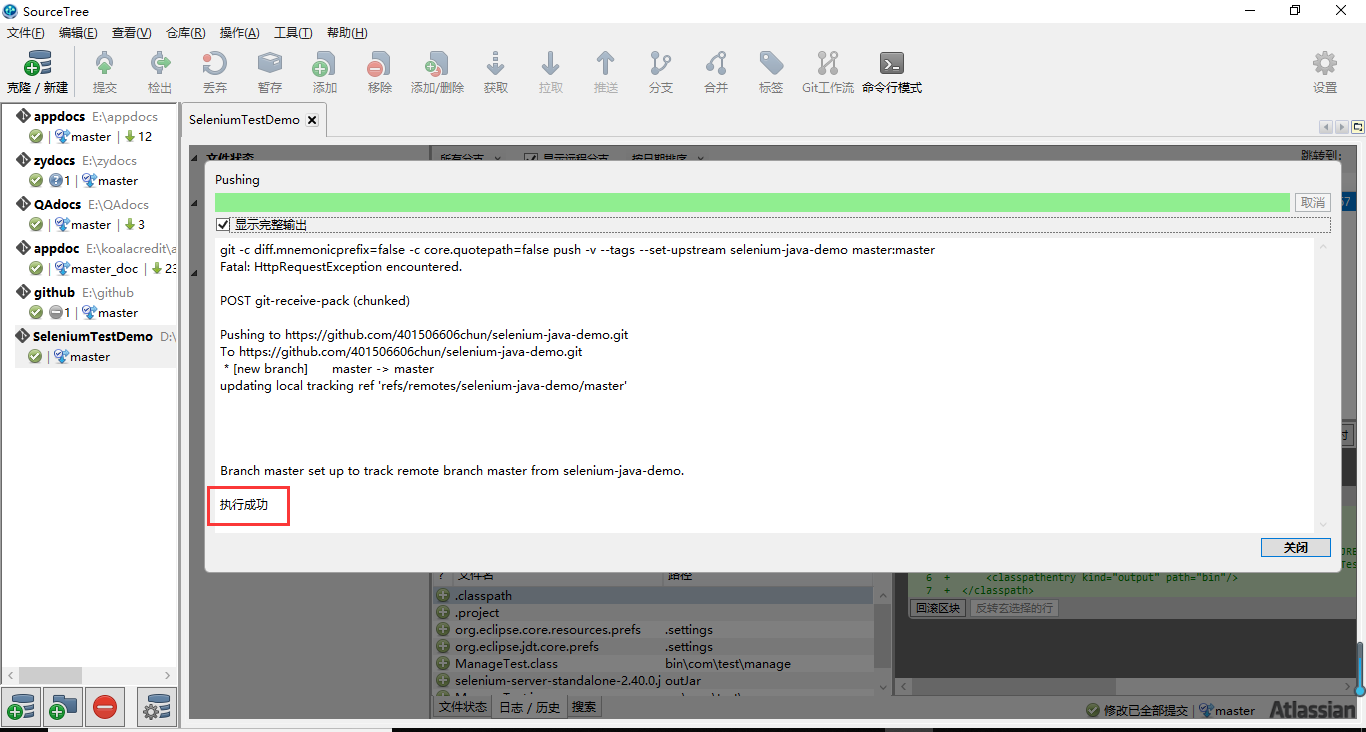




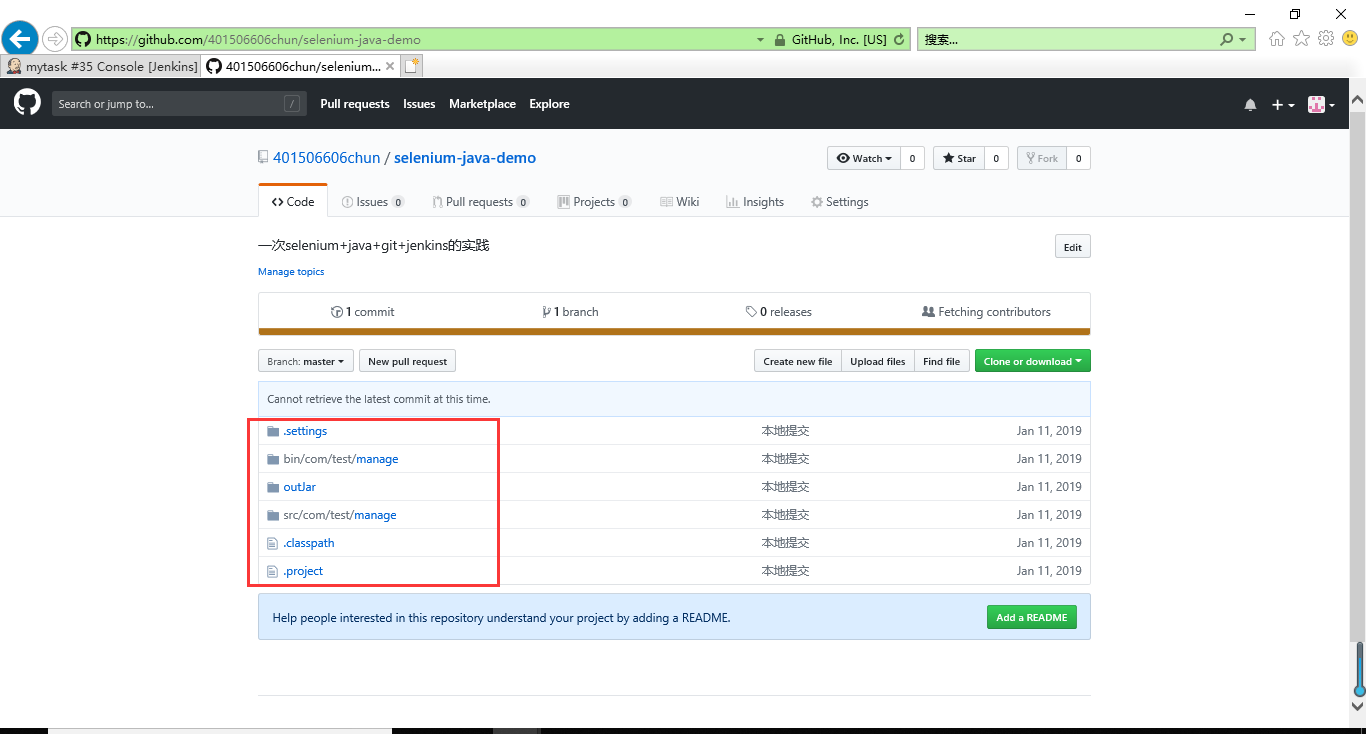


接下来需要进行推送





去github刷新查看最新提交的内容，此时已完成提交。



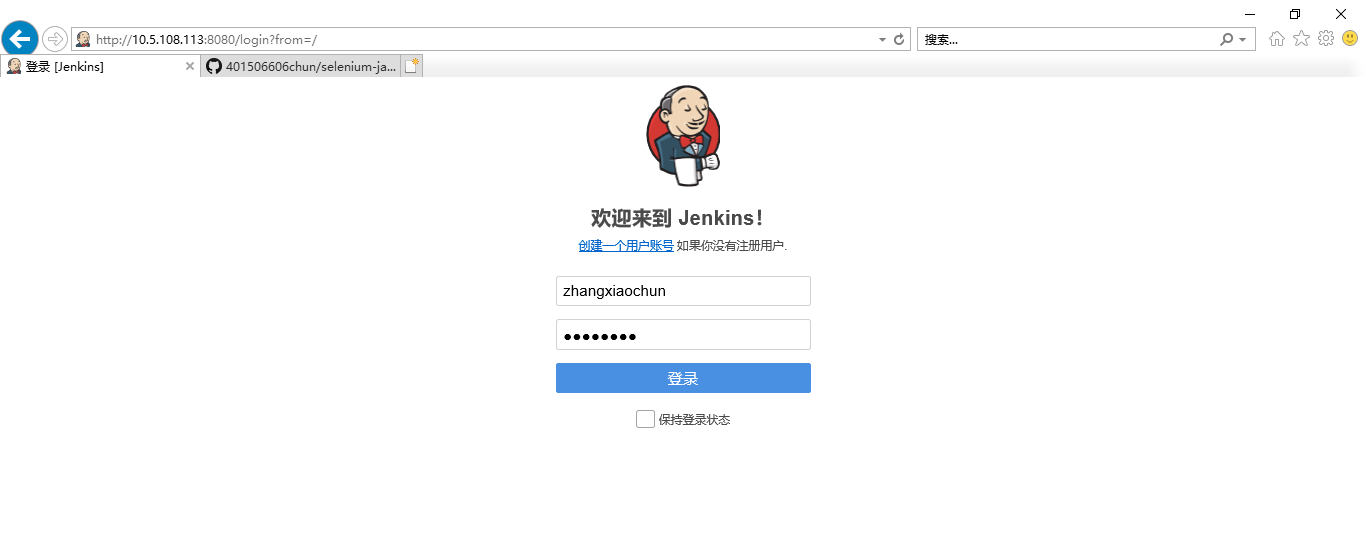
# 借用jenkins实现自动构建

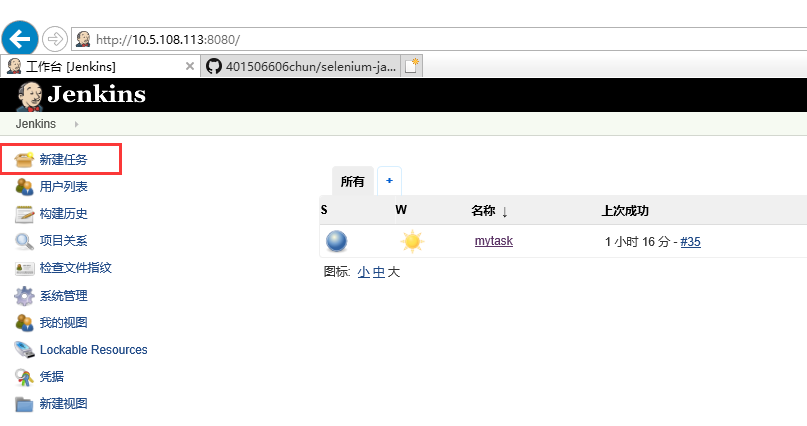
### **3.1jenkins环境搭建**

环境搭建不赘述

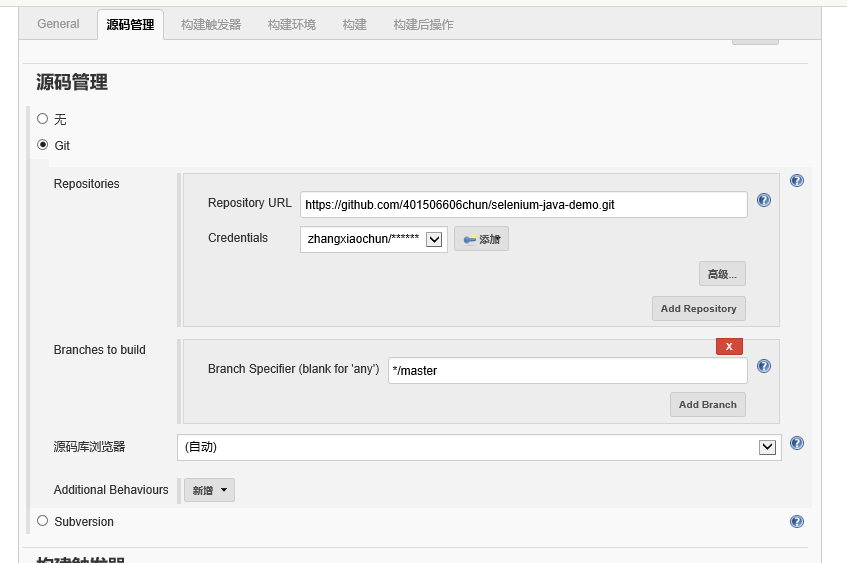
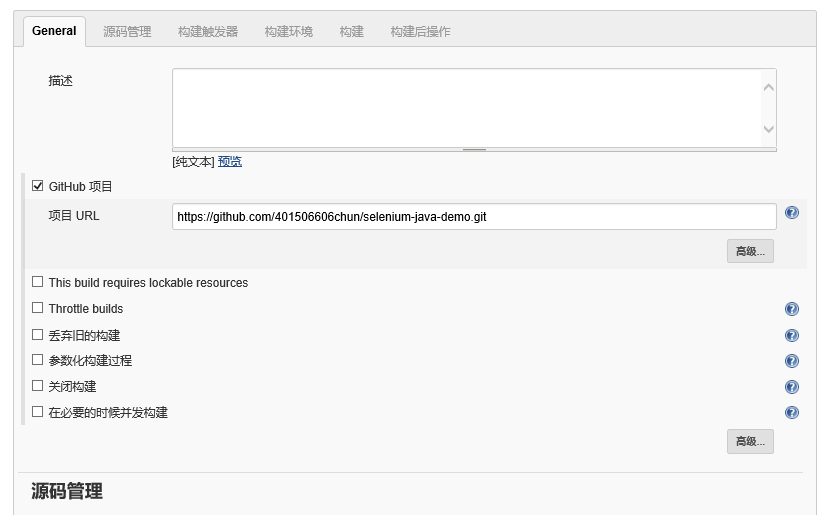
**3.2创建任务**

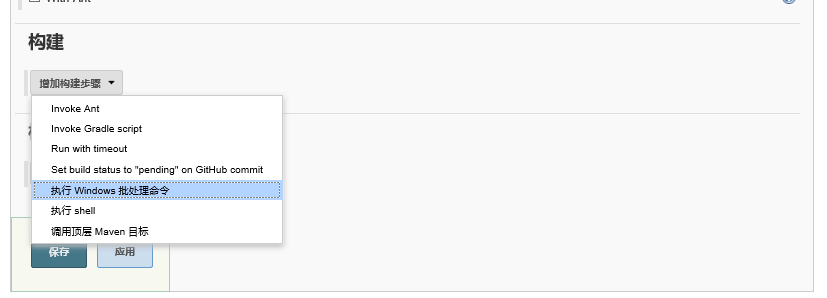
登录jenkins，并添加任务

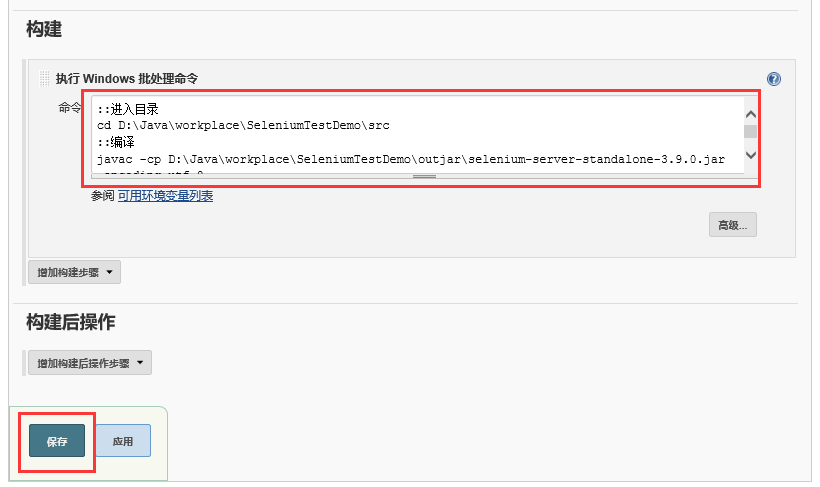










批处理命令具体如下：





至此任务创建成功了。

**3.3手动执行任务和定时构建**

#### 3.3.1手动如下：



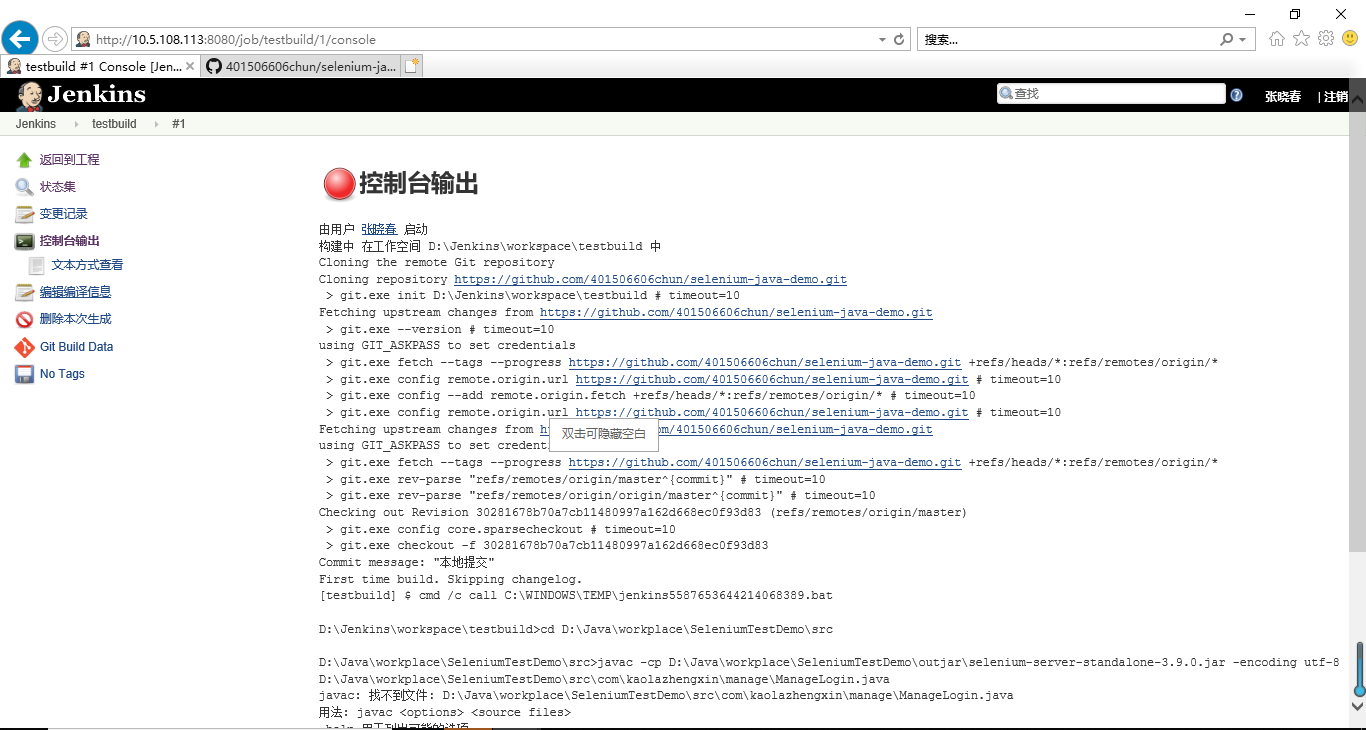


此处可以查看任务执行状态

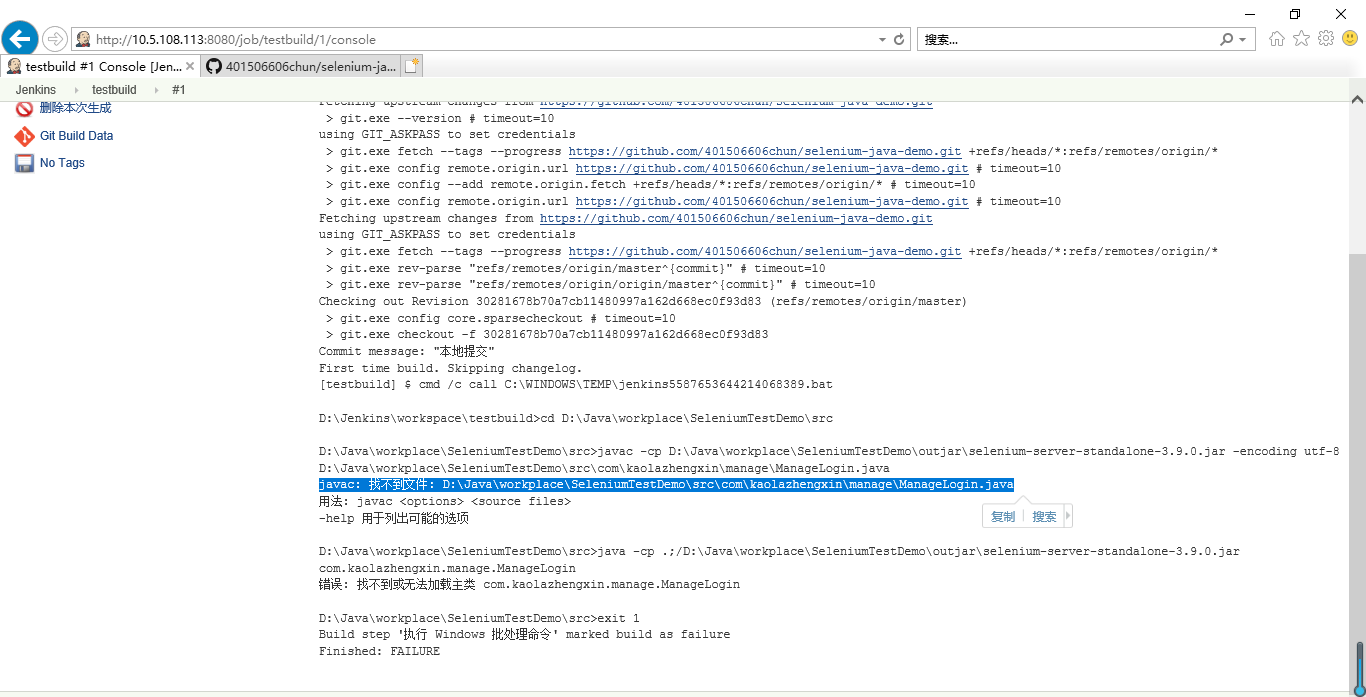
显示当前任务执行失败，我们可以进入控制台查看具体报错方便调试。



调试过程：



发现报错原因是javac编译报错：



修改任务配置中的批处理命令，并重新构建。



调试过程中遇到构建一直不结束的情况，最终替换jar包解决。

