****

**软件测试技术第二次实验报告**



**学 院 软件学院**

**专 业 软件工程**

**年 级 16级**

**姓 名 李今**

**2019年 3 月 31 日**

# 软件测试技术第一次实验报告

git URL

https://github.com/riginAI/test/tree/master/Lab2

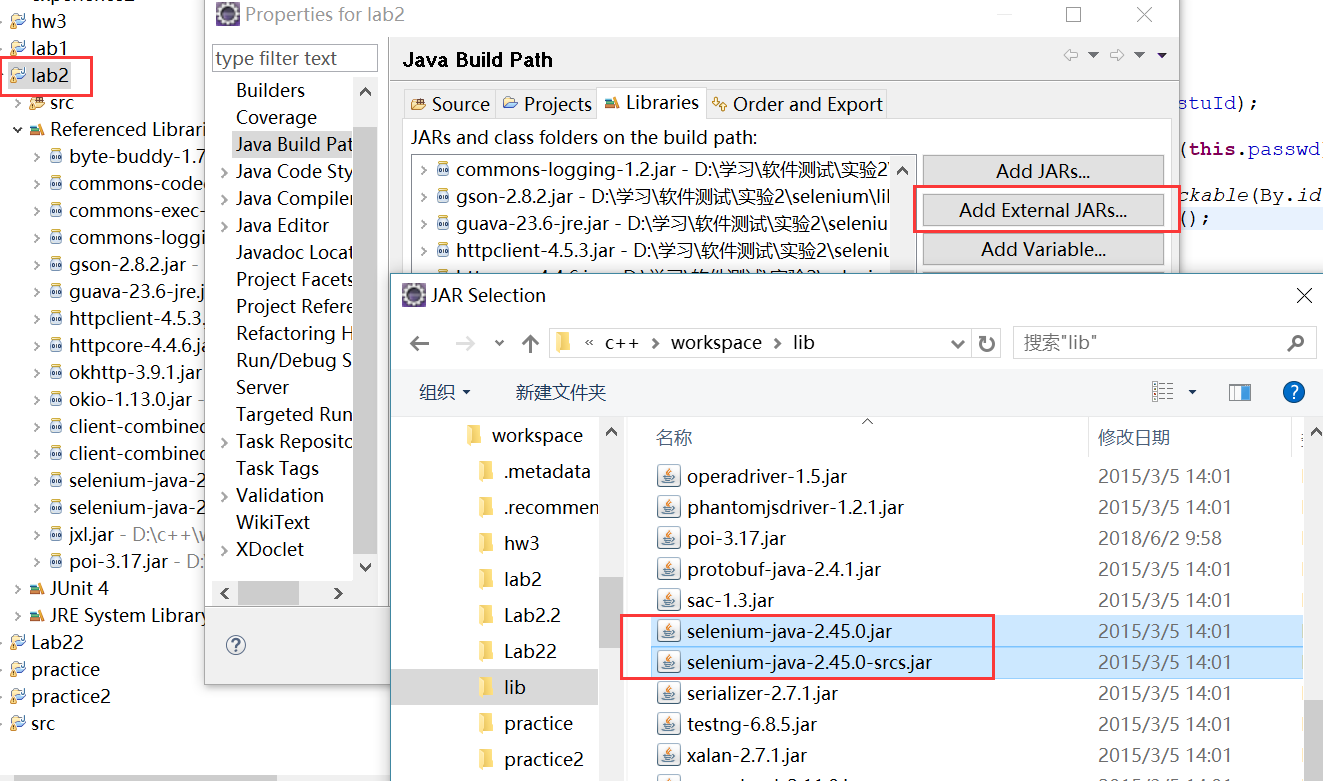
1. 需求分析

“软件测试名单.xlsx” contains information about the students, and <http://121.193.130.195:8800> can view someone’s information after logging in (student id as username, the last 6 digits of student id as password). Please check each record in the excel to make sure that each student’s information is consistent with the information on the website.

1. 环境配置
2. 安装selenium

由于在实验课上老师分享了selenium包，因此直接导入jar包即可，选择Build Path - Configure Build Path

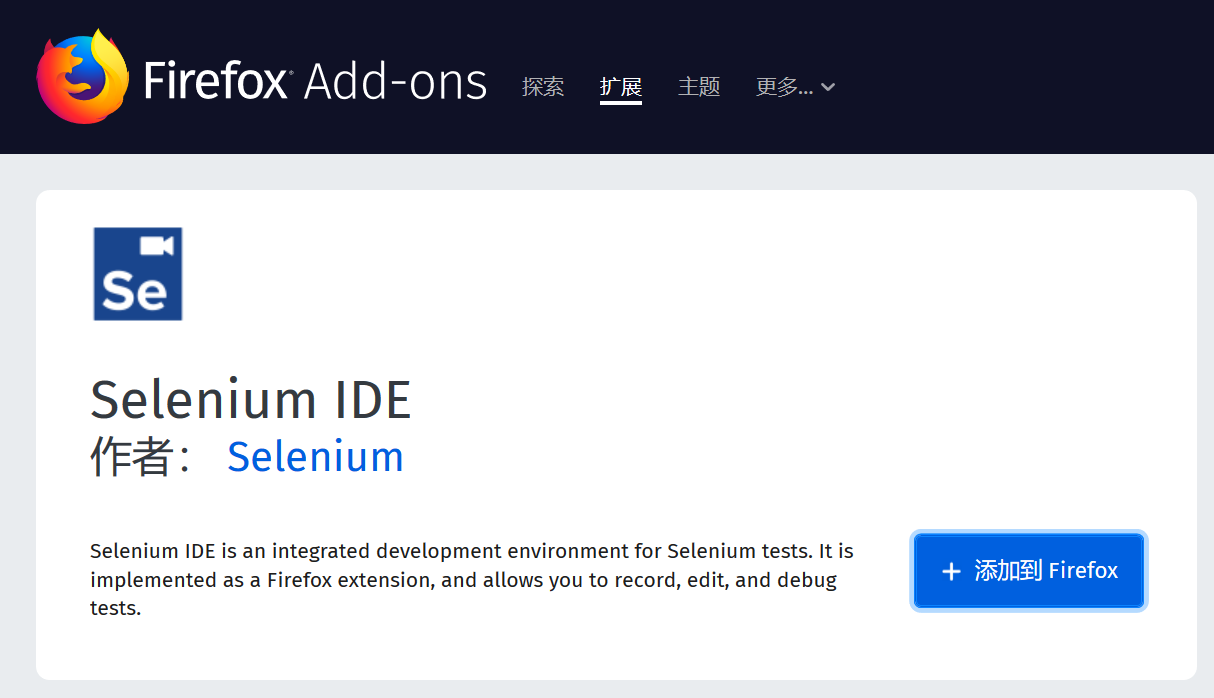
然后点击Libraries - Add External JARs，选择selenium的jar包，点击打开即可



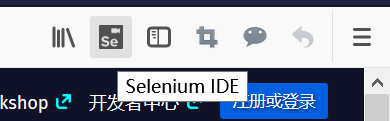
1. 安装Firefox 和 SeleniumIDE

Firefox直接在官网上下载最新版本即可

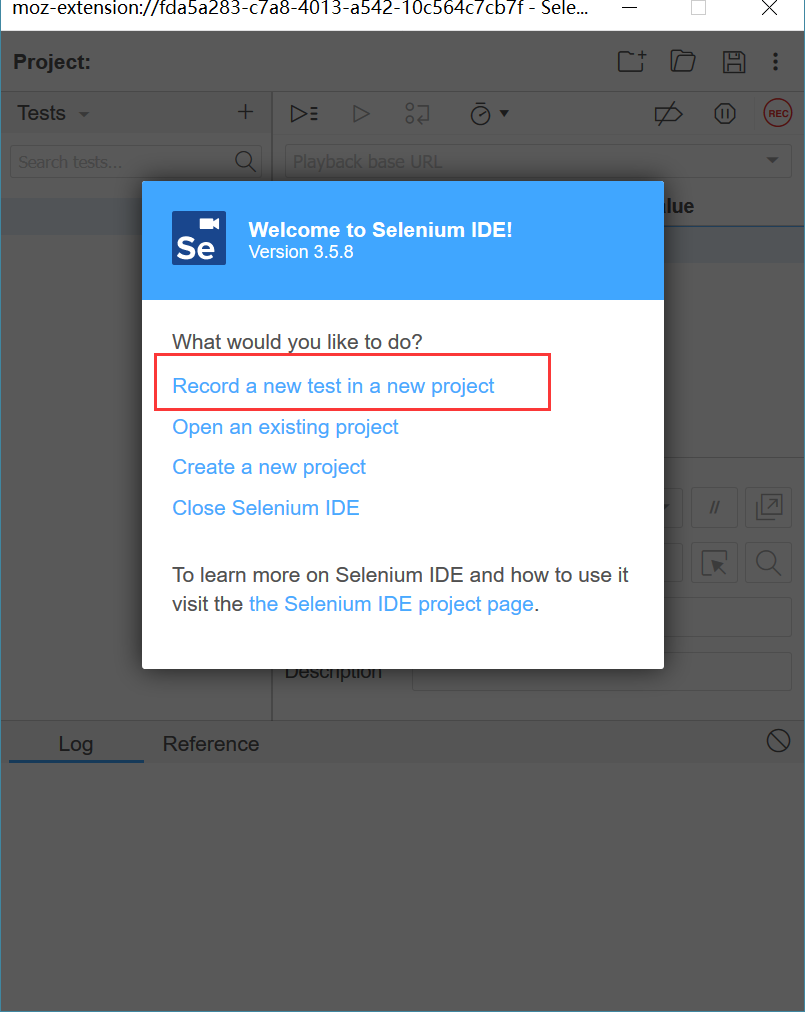
SeleniumIDE则在Firefox add-ones上搜索selenium，点击添加到Firefox



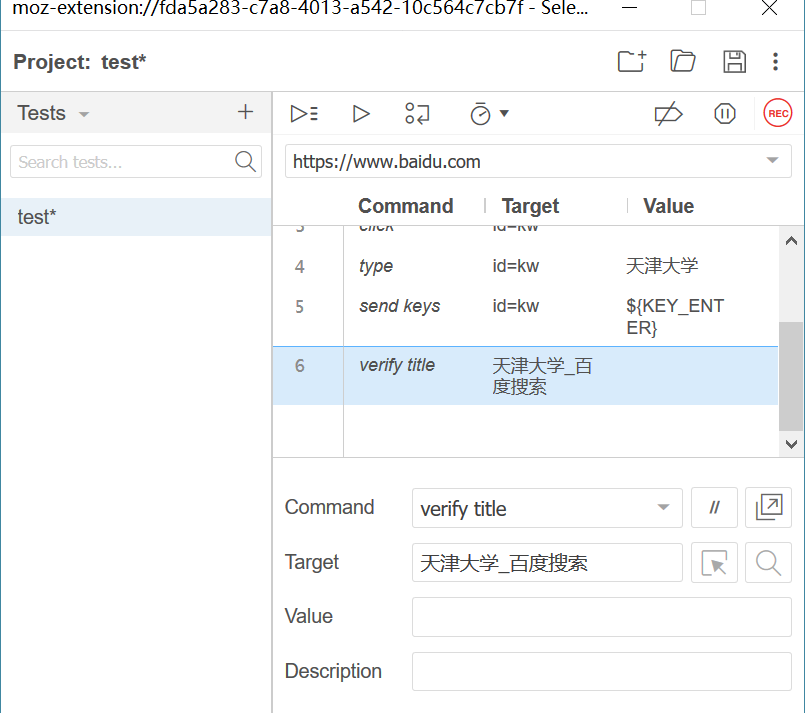
之后可以在右上角看到seleniumIDE的点击按钮



打开seleniumIDE

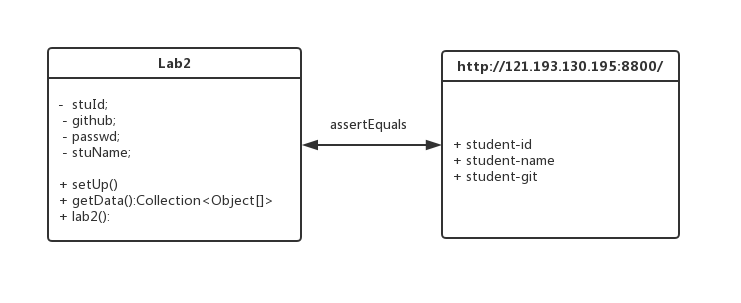


之后输入项目名，输入测试url即可开始录制



1. 概要设计

本次实验主要目的是测试网站数据，因此只设计了一个Lab2类，使用集成测试，来测试网站。



1. 详细设计

主要是@parameters部分和@test部分配合，遍历Excel文件对逐条数据进行测试

//部分核心代码：

@Parameters

**public** **static** Collection<Object[]> getData() **throws** BiffException, IOException {

File file = **new** File("D:/软件测试名单.xls");

Workbook book = Workbook.getWorkbook(file);

Sheet sheet = book.getSheet(0);

Object[][] content = **new** Object[142][4];

**for** (**int** i = 2; i < sheet.getRows(); i++) {

content[i-2][0] = sheet.getCell(1, i).getContents();//id

content[i-2][1] = sheet.getCell(1, i).getContents().substring(4);//password

content[i-2][2] = sheet.getCell(2, i).getContents();//name

content[i-2][3] = sheet.getCell(3, i).getContents();//gitUrl

}

List<Object[]> result = Arrays.asList(objects);

**return** result;

}

@Test

**public** **void** lab2() **throws** Exception {

driver.get(baseUrl + "/");

driver.findElement(By.name("id")).clear();

driver.findElement(By.name("id")).sendKeys(**this**.stuId);

driver.findElement(By.name("password")).clear();

driver.findElement(By.name("password")).sendKeys(**this**.passwd);

**try** {

wait.until(ExpectedConditions.elementToBeClickable(By.id("btn\_login")));

driver.findElement(By.id("btn\_login")).click();

} **catch** (Exception e) {

Thread.sleep(20000);

driver.navigate().refresh();

Thread.sleep(20000);

} **finally** {

//wait.until(ExpectedConditions.presenceOfElementLocated(By.id("student-id")));

WebElement checkId = driver.findElement(By.id("student-id"));

assertEquals(**this**.stuId, checkId.getText());

WebElement checkName = driver.findElement(By.id("student-name"));

assertEquals(**this**.stuName, checkName.getText());

WebElement checkUrl = driver.findElement(By.id("student-git"));

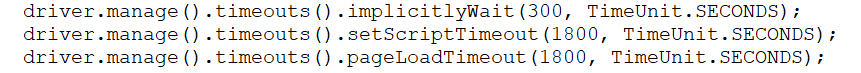
assertEquals(**this**.github, checkUrl.getText());

}

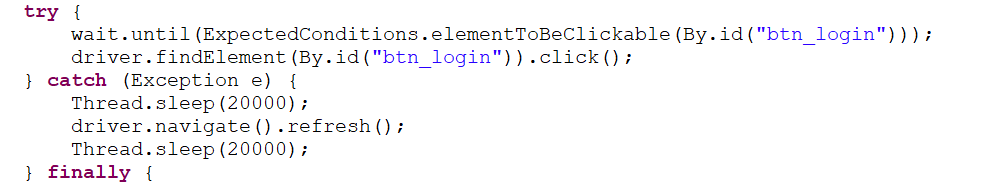
}

1. 调试分析

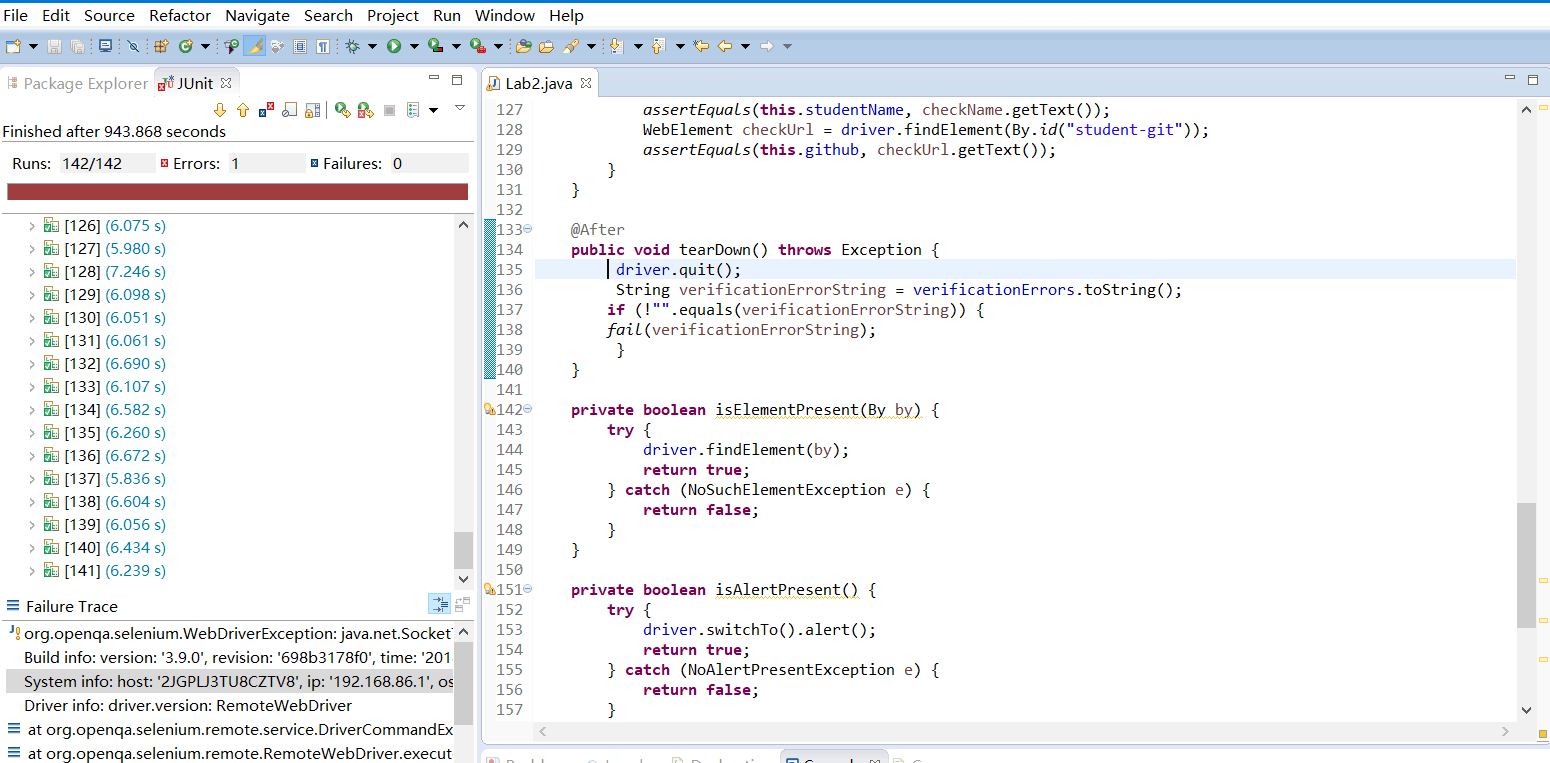
实验问题主要集中在会出现超时timeout exception上，为解决这个问题我上网查到了implicitlyWait等三个设置延时的方法，然而都没有解决问题

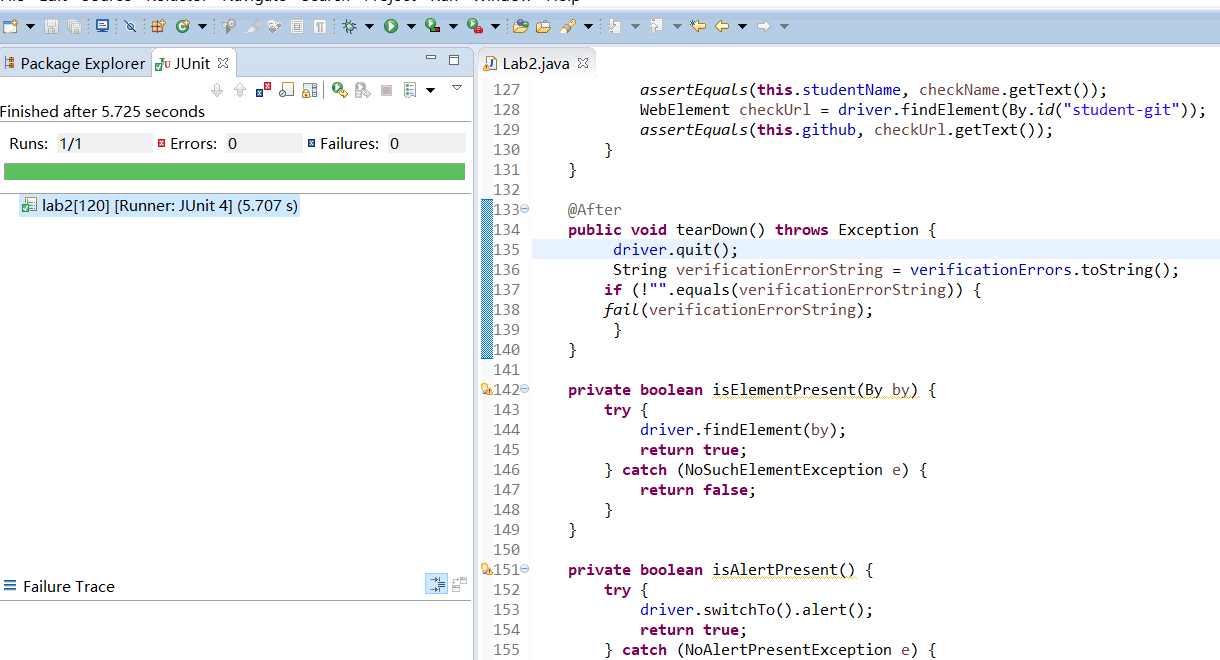


最后和同学交流后使用了显示等待WebDriverWait以及try catch，问题得以解决



1. 测试结果





可以看到尽管如此142条数据中第120条数据仍然出现了timeout exception，再次补测第120条数据后成功。

1. 总结

这次实验还是非常具有实际意义的。不管在之后的学习还是工作中，大量搜集信息、验证信息都是频繁出现的工作，而selenium作为一个用于Web应用程序测试的工具可以帮助我们实现页面测试自动化，快速获取网络信息，为我们节省大量的时间和精力，也让大量数据的测试提供了途径。