****

**软件测试技术第二次实验报告**



**学 院 智能与计算学部**

**专 业 软件工程**

**年 级 2016**

**姓 名 宋琦琪**

**2019年 3 月 31 日**

# 软件测试技术第二次实验报告

1. 需求分析（描述具体需求）
2. 在Eclipse中安装Selenium。
3. 安装Firefox和SeleniumIDE插件。
4. 尝试使用SeleniumIDE记录和导出脚本。
5. 请使用Selenium完成以下任务：

“软件测试名单.xlsx”包含有关学生的信息，软件测试名单可以在登录<http://121.193.130.195:8800> 后查看某人的信息(学生ID为用户名，学生ID的最后6位数字为密码)。请检查Excel中的每一条记录，以确保每个学生的信息与网站上的信息一致。

1. 概要设计（简单描述设计思路，配合UML图）

安装:

1. 在Eclipse中安装Selenium

(1)在eclipse中新建一个java project，并下载Selenium jar包

(2)选择新建的工程，将Selenium jar包中的lib文件夹和selenium-java-2.52.0.jar以及selenium-java-2.52.0-srcs.jar复制粘贴进去。

2. 安装Firefox和SeleniumIDE插件

(1)下载火狐浏览器并按步骤安装

(2)百度搜索Add-ons Firefox并进入页面，下载Selenium插件

代码:

1. 在java项目中导入实验要求的jar包（包括junit相关jar包），下载gekcodriver.exe并导入src下。
2. 使用katalon recorder，点击record新建并录制脚本，点击stop完成记录。点击export，并选择Java+unit，导出相应脚本。
3. 在eclipse中对导出的代码进行修改，加入csv读取、网页文本元素获取、url判断等相应代码。
4. 详细设计（详细描述具体如何实现，附代码及说明）

**package** cn.tjucic.selenium;

**import** java.util.regex.Pattern;

**import** java.io.File;

**import** java.nio.charset.Charset;

**import** java.util.concurrent.TimeUnit;

**import** org.junit.\*;

**import** **static** org.junit.Assert.\*;

**import** **static** org.hamcrest.CoreMatchers.\*;

**import** org.openqa.selenium.\*;

**import** org.openqa.selenium.firefox.FirefoxDriver;

**import** org.openqa.selenium.support.ui.Select;

**import** com.csvreader.CsvReader;

**public** **class** TestLab2 {

**private** WebDriver driver;

**private** String baseUrl;

**private** **boolean** acceptNextAlert = **true**;

**private** StringBuffer verificationErrors = **new** StringBuffer();

**private** String id = **null**;

**private** String pwd = **null**;

**private** String gitUrl = **null**;

@Before

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

String driverPath = System.*getProperty*("user.dir") + "/src/geckodriver.exe";

System.*setProperty*("webdriver.gecko.driver", driverPath);

driver = **new** FirefoxDriver();

baseUrl = "http://121.193.130.195:8800";

driver.manage().timeouts().implicitlyWait(30, TimeUnit.***SECONDS***);

}

@Test

**public** **void** testLab2() **throws** Exception {

CsvReader cin = **new** CsvReader("F:/eclipse/2.Lab/test.csv", ',', Charset.*forName*("GBK"));

//读取csv文件

cin.readHeaders();

**while** (cin.readRecord()) {

id = cin.get(1);//获得学号（ID）

**if**(id.length()<10){

**continue**;

}

pwd = id.substring(4, 10);//获得密码

gitUrl = cin.get(3);//获得表格中的url

driver.get("http://121.193.130.195:8800/login");

WebElement we = driver.findElement(By.*name*("id"));

we.click();

// driver.findElement(By.name("id")).click();

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

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

// driver.findElement(By.name("password")).click();

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

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

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

String gitUrls = driver.findElement(By.*xpath*("//\*[@id='student-git']")).getText();

//获得网页中url文本 使用xpath finder定位

**if** (!gitUrl.equals(gitUrls)) {

System.***out***.println(id);

System.***out***.println(gitUrl);

System.***out***.println(gitUrls);

**continue**;

}//判断网页中的url和表格中的url是否相同，相同则继续，不同则输出

*assertEquals*(gitUrl, gitUrls);

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

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

}

}

@After

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

// driver.quit();

// String verificationErrorString = verificationErrors.toString();

// if (!"".equals(verificationErrorString)) {

// fail(verificationErrorString);

// }

}

**private** **boolean** isElementPresent(By by) {

**try** {

driver.findElement(by);

**return** **true**;

} **catch** (NoSuchElementException e) {

**return** **false**;

}

}

**private** **boolean** isAlertPresent() {

**try** {

driver.switchTo().alert();

**return** **true**;

} **catch** (NoAlertPresentException e) {

**return** **false**;

}

}

**private** String closeAlertAndGetItsText() {

**try** {

Alert alert = driver.switchTo().alert();

String alertText = alert.getText();

**if** (acceptNextAlert) {

alert.accept();

} **else** {

alert.dismiss();

}

**return** alertText;

} **finally** {

acceptNextAlert = **true**;

}

}

}

1. 调试分析（在实验过程中遇到的问题以及如何解决）
2. 问题:不会使用selenium导出录制好的项目代码

解决：通过网上查询得知，新版的selenium没有导出功能，所以下载了功能相同的katalon recorder录制后导出Java代码

1. 问题：经历了很多次timeout报错，而且手机热点无法登陆实验要求的网站

解决：不断修改等待时间，但还是会偶尔出现超时的情况

1. 问题：在读取csv文件的时候，第一次路径写的是带有中文的绝对路径，出现了csv文件does not exist的情况；将csv文件放在java项目下使用相对路径的时候，还是同样的报错。

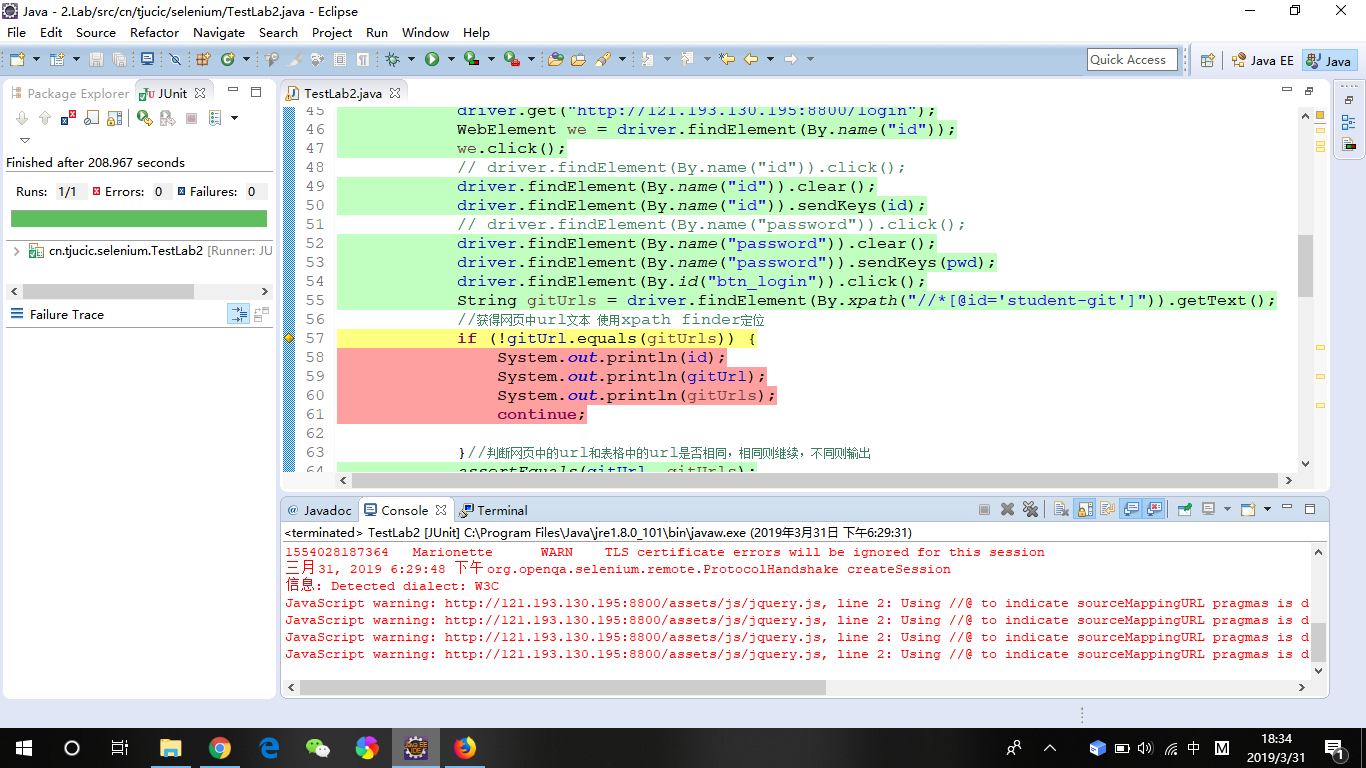
解决：在网上查找原因时发现java读取带有中文的路径时会出现乱码，放入项目下使用相对路径的原因还不清楚，但改成绝对路径后可以运行。

1. 问题：不知道如何使用selenium获得网页中的文本信息

解决：通过百度查询得知可以使用xpath获取文本，通过火狐add-ons下载了xpath finder插件，定位了url文本元素所在位置，得以完成xpath语法部分。

1. 测试结果（描述输入和输出）

无输出（说明表格与网页中的url全部一致）



1. 总结

由于网速问题，本次实验经历了从一开始不知道出现错误的原因，后来通过同学的提醒修改等待时间、更换网络等方法，暂时解决了超时问题，但是在程序运行中还有一些因为不当操作或者不熟悉语言导致的错误，但是junit的使用让我准确的定位了错误出现的位置，此次实验使用selenium录制网页操作，让我觉得十分新奇有趣，但是在修改代码的过程中我也感受到了它的复杂。本次实验也让我意识到了软件测试在实际应用中的重要性。