****

**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级**

**姓 名 孙玮**

**2019年 3月 26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）

1.使用Eclipse安装Selenium。

2.安装Firefox和SeleniumIDE插件。

3.尝试使用SeleniumIDE录制和导出脚本。

4.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，并且网页http://121.193.130.195:8800可以在登录后查看某人的信息（学生ID为用户名，学生ID的最后6位为密码）。 请检查excel中的每条记录，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）
2. 使用Eclipse安装Selenium：
3. 从官网上下载Selenium的jar包
4. 建立Java项目，将jar包添加到相应的构建路径中
5. 安装Firefox和SeleniumIDE插件：
6. 运行从官网下载的Firefox在线安装包，并安装到默认路径。
7. 在firefox扩展中安装我们想要的SeleniumIDE插件
8. 尝试使用SeleniumIDE录制和导出脚本：
9. 安装SeleniumIDE后，在浏览器右上角就会有相应的标志按钮; 点击此按钮就会出现SeleniumIDE操作界面; 打开一个任意网页之后，打开seleniumIDE使其保持录制状态，就可以操作了，完成操作之后，对其停止，脚本录制完成。
10. 按照下列顺序点击操作，文件→ETCA→JAVA；把文件保存在适当位置，命名；此时可以用notepad++等编辑软件打开文件编辑了。
11. 对于校验学生名单问题的概要设计：

设计思路：读取“软件测试名单.xlsx”excel表中的内容，并保存在一个数组中，然后通过循环的方式利用Selenium进行校对。

1. 详细设计（详细描述具体如何实现，附代码及说明）

@Test

public void testBaidu() throws Exception {

TestXlsx test = new TestXlsx();

//读取excel数据并存入二维数组

test.getarray();

String a[][] = new String[145][4];

System.arraycopy(test.getarray(),0,a,0,test.getarray().length);

driver.get(baseUrl + "/");

//逐一进行比较

for(int i = 2; i < 145; i++) {

WebElement input\_name = driver.findElement(By.name("id"));

input\_name.clear();

input\_name.sendKeys(a[i][1]);

WebElement input\_pwd = driver.findElement(By.name("password"));

input\_pwd.clear();

input\_pwd.sendKeys(a[i][1].substring(4));

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

btn.click();

assertEquals(a[i][1], driver.findElement(By.id("student-id")).getText());

assertEquals(a[i][2], driver.findElement(By.id("student-name")).getText());

assertEquals(a[i][3], driver.findElement(By.id("student-git")).getText());

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

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

}

}

1. 调试分析（在实验过程中遇到的问题以及如何解决）

学号尾号为164的信息在一开始测试过程中总会出现崩溃，最后比对了网页源码和excel表的内容，发现excel的git地址一栏多了一个空格。进行了trim()等的处理，解决了该问题。

1. 测试结果（描述输入和输出）

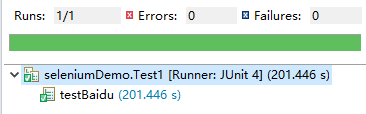
测试样例及输入

assertEquals(arr[k][0], driver.findElement(By.id("student-id")).getText());

assertEquals(arr[k][1], driver.findElement(By.id("student-name")).getText());

assertEquals(arr[k][2], driver.findElement(By.id("student-git")).getText());

输出：



coverage report



1. 总结

软件测试工作是一个系统而复杂的工程，通过本次实验，我学会了利用Selenium进行浏览器自动化测试。并通过录制导出代码，了解了相关代码方面的操作。我希望在打好基础的同时，在接下来的实验中，尽一切可能的去模拟、去思考现实使用环境中的应用可能性和预测可能出现的状况，以更好地实现测试的效果和作用。进一步体会到了Selenium自动测试化框架的方便高效。

Github: https://github.com/Vincent-Sun/software-test/lab2