****

**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 傅淏**

**2019年 3 月 30 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. Install Selenium with Eclipse.

2. Install Firefox and SeleniumIDE plugin.

3. Try to record and export scripts using SeleniumIDE.

4. Please complete the following task using Selenium:

“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. 概要设计

**设计思路：**

1、导入POI包，使用workbook将Excel文件的数据读取到字符串中；

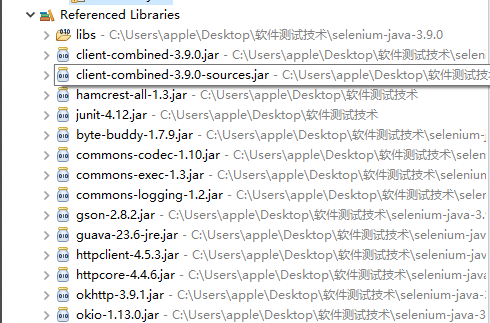
2、利用selenium和katalon recorder在火狐浏览器中生成基本的测试文件；

3、编写JAVA程序自动使用读取的Excel数据重复登录测试网站；

4、将Excel中的数据与网站中获取的数据进行比对，重复过程直至所有数据对比完毕。

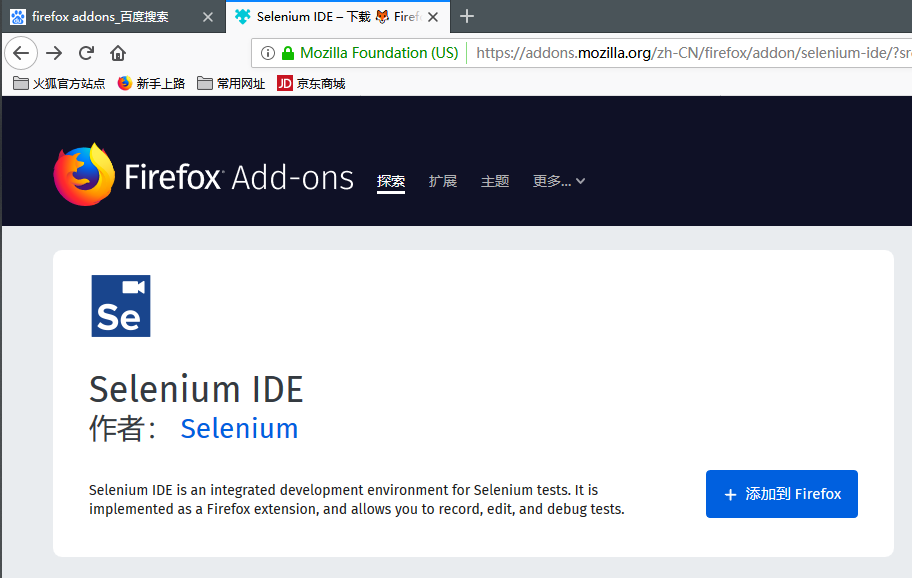
1. 详细设计（详细描述具体如何实现，附代码及说明）

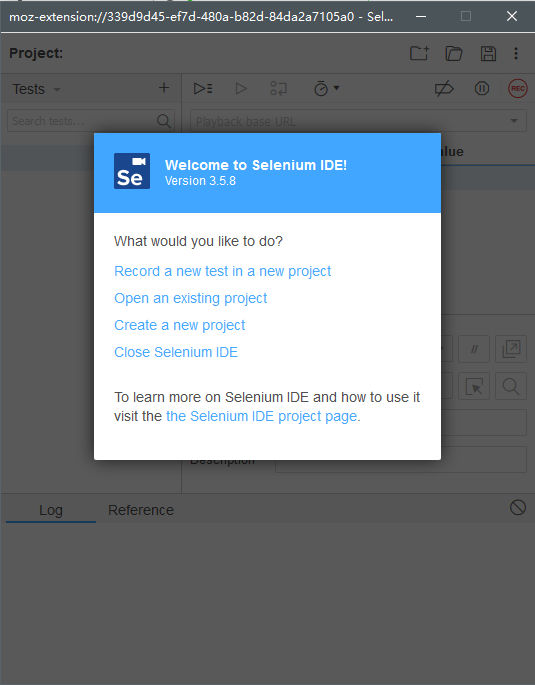
1、在eclipse上设置安装selenium，导入所有jar包；

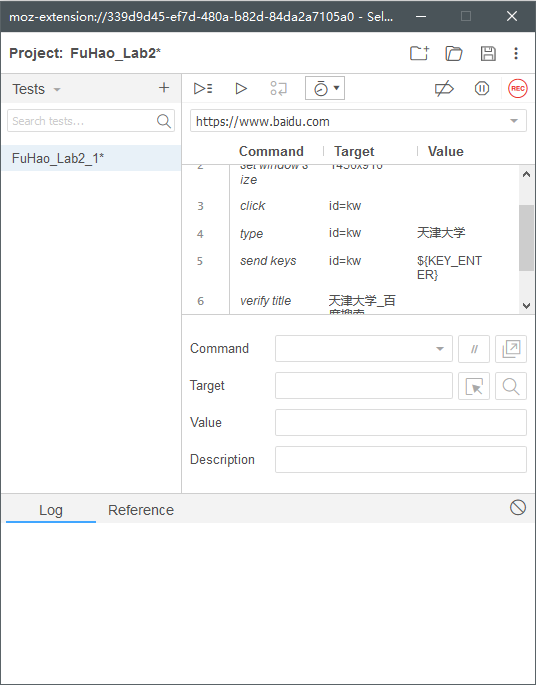


2、安装火狐浏览器，并在其中安装selenium IDE插件，进行简单的网站测试；

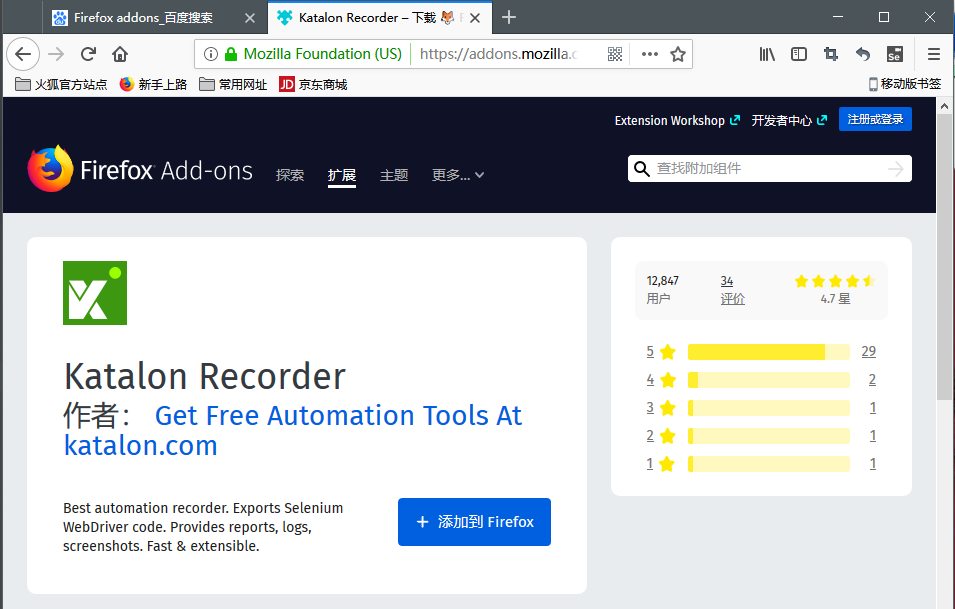


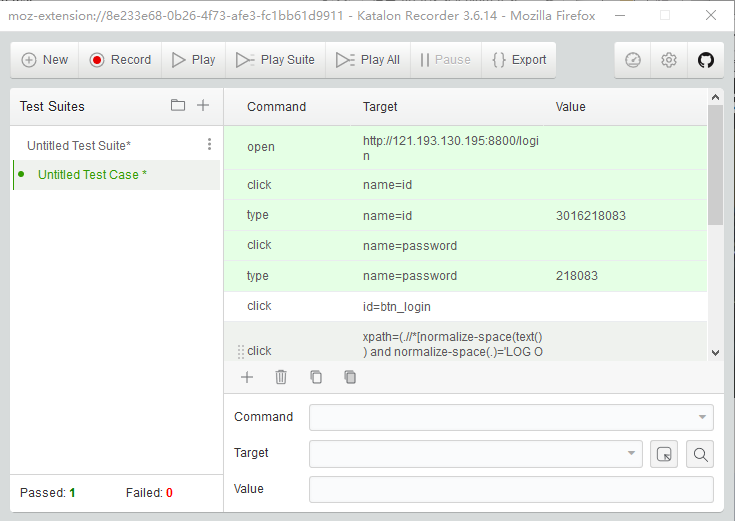






3、使用katalon recorder测试实验网站，并生成JAVA测试文件；





4、编写代码

4.1 Excel.java

package scs\_lab2;

import java.io.File;

import java.io.FileInputStream;

import java.io.InputStream;

import org.apache.poi.ss.usermodel.Cell;

import org.apache.poi.ss.usermodel.CellType;

import org.apache.poi.ss.usermodel.Row;

import org.apache.poi.ss.usermodel.Sheet;

import org.apache.poi.xssf.usermodel.XSSFWorkbook;

public class Excel {

public static String getExcelData(){

File file = new File(System.getProperty("user.dir") + "/firefox/list.xlsx");

InputStream inputStream = null;

XSSFWorkbook workbook = null;

String str = "";

try {

inputStream = new FileInputStream(file);

workbook = new XSSFWorkbook(inputStream);

inputStream.close();

//工作表对象

Sheet sheet = workbook.getSheetAt(0);

//总行数

int rowLength = sheet.getLastRowNum()+1;

//工作表的列

Row row = sheet.getRow(2);

//总列数

int colLength = row.getLastCellNum();

//得到指定的单元格

Cell cell = row.getCell(1);

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

row = sheet.getRow(i);

for (int j = 1; j < colLength; j++) {

cell = row.getCell(j);

if (cell != null)

cell.setCellType(CellType.STRING);

str = str + cell.getStringCellValue() + "\t";

}

str = str + "\n";

}

} catch (Exception e) {

e.printStackTrace();

}

return str;

}

}

4.2 TestExcel.java

package scs\_lab2;

import java.util.regex.Pattern;

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;

public class TestExcel {

private WebDriver driver;

private String baseUrl;

private boolean acceptNextAlert = true;

private StringBuffer verificationErrors = new StringBuffer();

@Before

public void setUp() throws Exception {

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

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

driver = new FirefoxDriver();

baseUrl = "https://www.baidu.com/";

driver.manage().timeouts().implicitlyWait(300, TimeUnit.SECONDS);

}

@Test

public void testExcel() throws Exception {

String excelDataStr = Excel.getExcelData();

WebElement button;

WebElement webGithubURL;

WebElement studentId;

WebElement studentName;

String[] everyoneData = excelDataStr.split("\n");

String someoneID = "";

String someoneName = "";

String someoneGitURL = "";

//进入测试网站

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

//截取字符串，获取各部分的数据进行测试比对

for(int i = 0, len = everyoneData.length ; i < len ; i++){

//对每个同学的不同信息数据进一步划分，存入对应的临时信息字符串

String[] someoneData = everyoneData[i].split("\t");

for(int j = 0, len2 = someoneData.length;j < len2; j++){

if(j == 0)

someoneID = someoneData[j];

else if(j == 1)

someoneName = someoneData[j];

else if(j == 2)

someoneGitURL = someoneData[j];

}

//利用selenium进行自动化测试

//截取密码（后六位）

String password = someoneID.substring(4);

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

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

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

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

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

button.submit();

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

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

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

assertEquals(someoneID, studentId.getText());

assertEquals(someoneName, studentName.getText());

assertEquals(someoneGitURL, webGithubURL.getText());

driver.findElement(By.id("btn\_logout")).sendKeys(Keys.ENTER);

driver.findElement(By.id("btn\_return")).sendKeys(Keys.ENTER);

someoneID = "";

someoneName = "";

someoneGitURL = "";

}

}

@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;

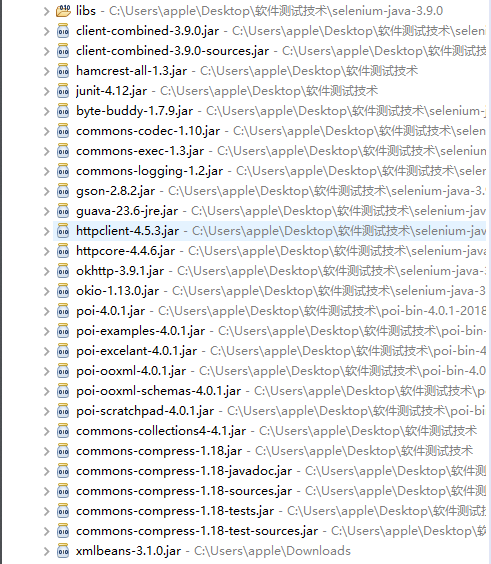
}

}

}

1. 调试分析（在实验过程中遇到的问题以及如何解决）

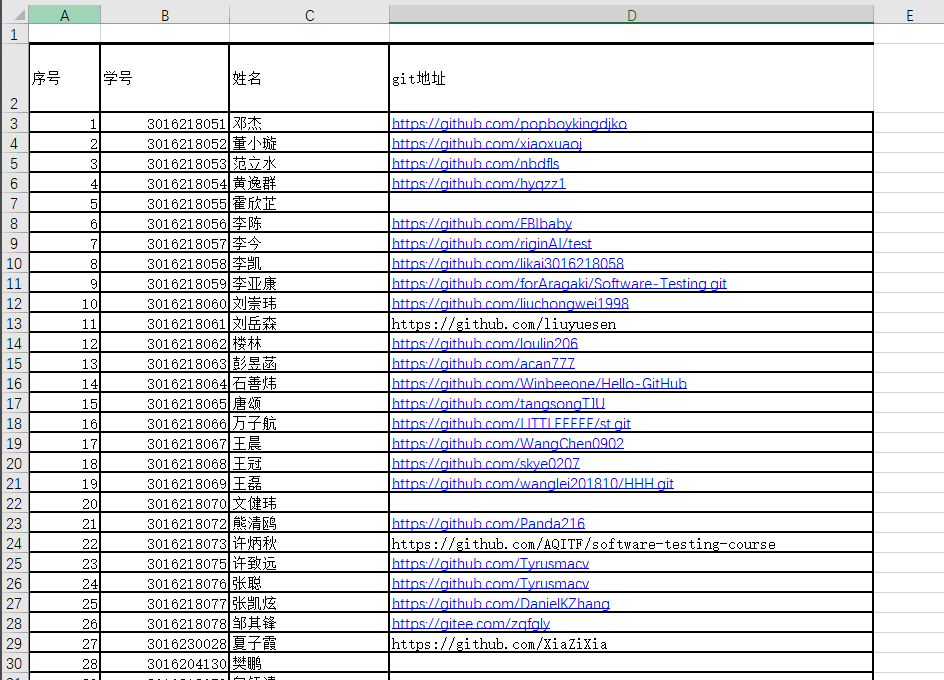
1、各种方法找不到，通过比对方法下载添加更多的jar包解决问题，最后所有所需的jar包均包含；



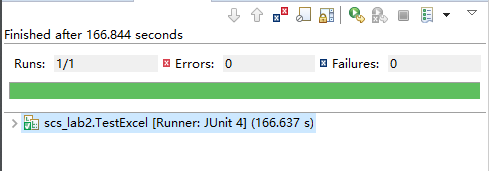
2、进行测试时火狐浏览器一直闪退，一开始以为是读取Excel文件读取不到，后来发现是使用的xmlbeans的jar包版本过低导致无法识别最新的Excel 2019，所以找到最新的xmlbeans-3.1.0这个jar包解决问题。

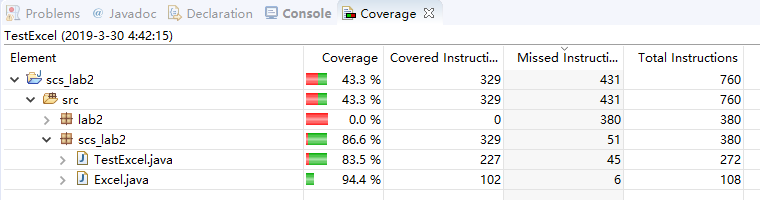
1. 测试结果（描述输入和输出）

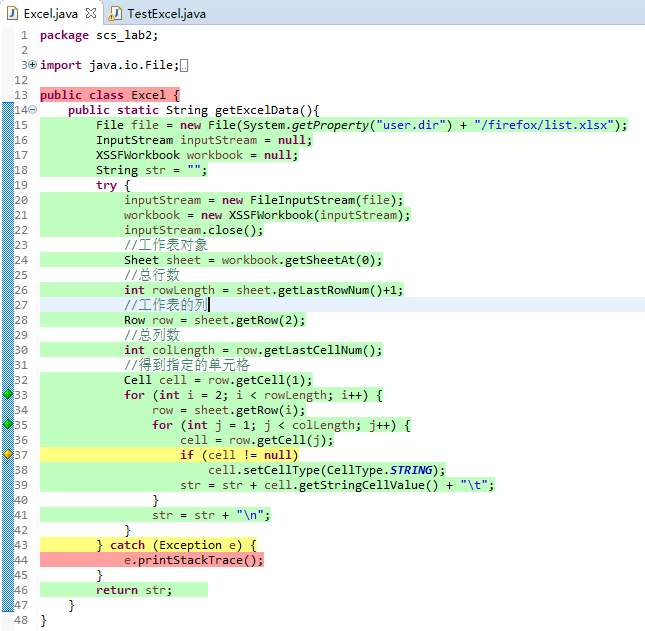
输入测试用例：

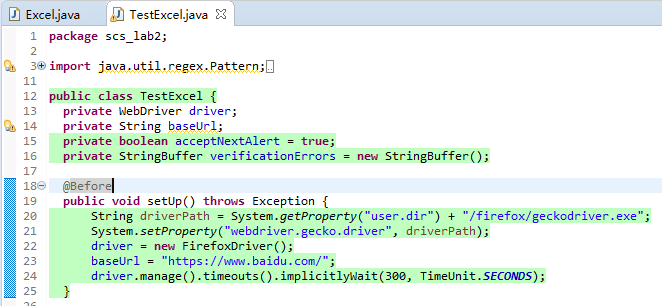


输出结果：









1. 总结

通过这次实验，我学会了使用selenium进行自动化测试，使用一些简单的脚本录制测试程序进行网页的测试；掌握了POI包在编程过程中的使用，使用eclipse读取Excel文件并进行数据编辑的方法；最后对于处理编程错误的熟练程度有了进一步的提高。

七、GitHub的URL

https://github.com/HaoFuTju