****

**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 王晨**

**2019年 4 月 1 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. Install Selenium with Eclipse.
3. Install Firefox and SeleniumIDE plugin.
4. Try to record and export scripts using SeleniumIDE.
5. 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（简单描述设计思路，配合UML图）

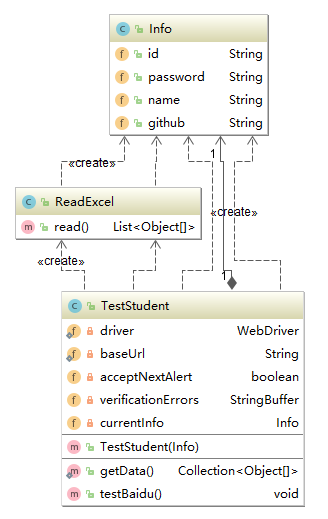
本次实验共分为3个类：

1.储存学生信息的数据类Info，

2.读取表格信息的操作类ReadExcel，

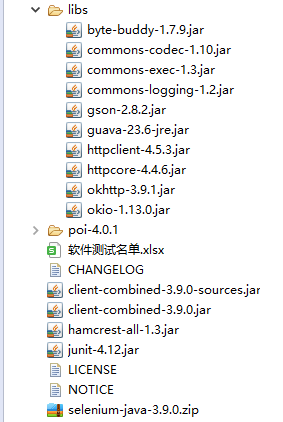
3.测试类TestStudent.

4.UML类图

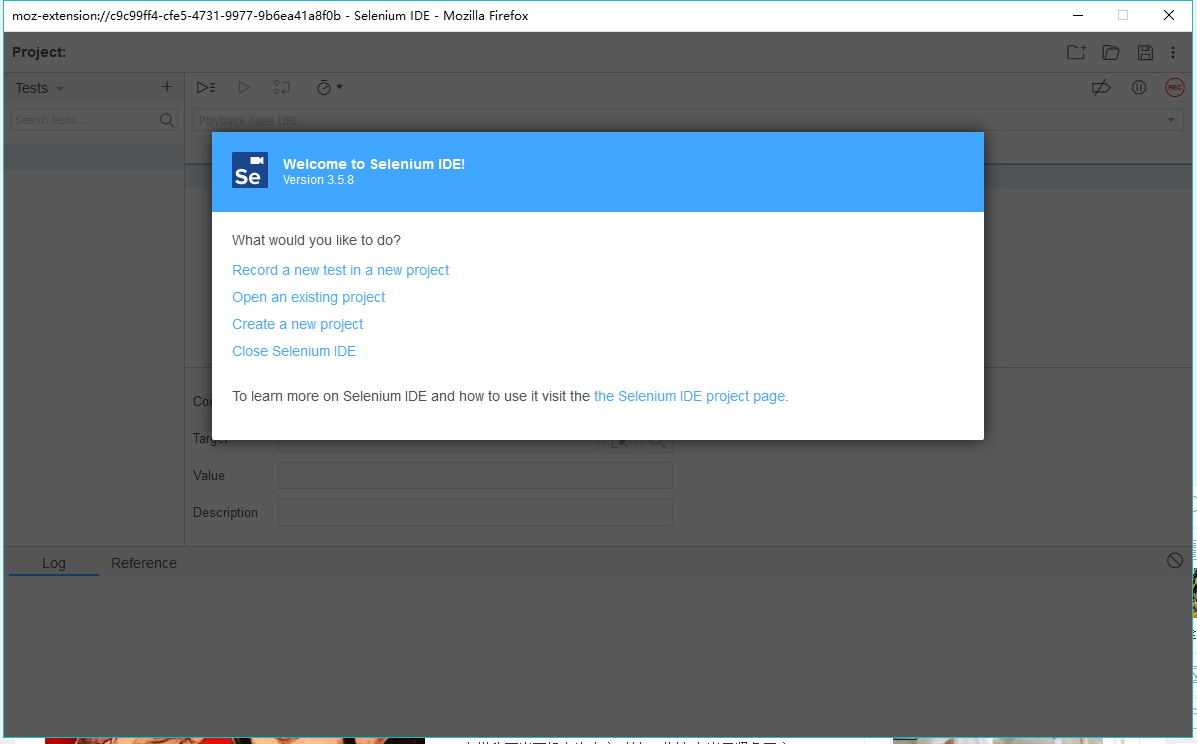


1. 详细设计（详细描述具体如何实现，附代码及说明）

1.安装Selenium with Eclipse



2.安装Selenium IDE with Firefox



2.Info类

package cn.tjucic.selenium;

public class Info {

public String id; //学生学号

public String password; //密码

public String name; //姓名

public String github; //github url

}

3.ReadExcel类

package cn.tjucic.selenium;

import java.io.File;

import java.util.ArrayList;

import java.util.List;

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.ss.usermodel.Workbook;

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

// 读取excel的类

public class ReadExcel {

public List<Object[]> read() throws Exception {

String filePath = System.getProperty("user.dir") + "\\软件测试名单.xlsx";

System.out.println(filePath);

File file = new File(filePath);

System.out.println(file);

List<Object[]> result = new ArrayList<Object[]>();

Workbook workbook = new XSSFWorkbook(file);

Sheet sheet = workbook.getSheetAt(0);

for(int rowIndex=2; rowIndex<=sheet.getPhysicalNumberOfRows(); rowIndex++) {

Row row = sheet.getRow(rowIndex);

if(row != null) {

Object[] infoList = new Info[1];

Info info = new Info();

info.id = String.valueOf((long)row.getCell(1).getNumericCellValue());

info.name = row.getCell(2).getStringCellValue();

info.github = row.getCell(3).getStringCellValue();

info.password = info.id.substring(4);

infoList[0] = info;

result.add(infoList);

}

}

System.out.println("read");

return result;

}

}

4.TestStudent类

package cn.tjucic.selenium;

import java.util.regex.Pattern;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.Collection;

import java.util.List;

import java.util.concurrent.TimeUnit;

import org.apache.poi.ss.formula.functions.Count;

import org.junit.\*;

import org.junit.runner.RunWith;

import org.junit.runners.Parameterized;

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 net.bytebuddy.asm.Advice.This;

@RunWith(Parameterized.class)

public class TestStudent {

private static WebDriver driver;

private static String baseUrl;

private boolean acceptNextAlert = true;

private StringBuffer verificationErrors = new StringBuffer();

private Info currentInfo = new Info();

public TestStudent(Info info) {

this.currentInfo = info;

}

@Parameterized.Parameters

public static Collection<Object[]> getData() 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(900, TimeUnit.SECONDS);

ReadExcel readExcel = new ReadExcel();

System.out.println(".");

List<Object[]> list = readExcel.read();

return list;

}

@Test

public void testBaidu() throws Exception {

driver.get(baseUrl);

// 输入内容，并确定

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

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

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

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

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

// 测试信息

assertEquals(currentInfo.id, driver.findElement(By.id("student-id")).getText());

assertEquals(currentInfo.name, driver.findElement(By.id("student-name")).getText());

assertEquals(currentInfo.github, driver.findElement(By.id("student-git")).getText());

// 返回

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

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

}

}

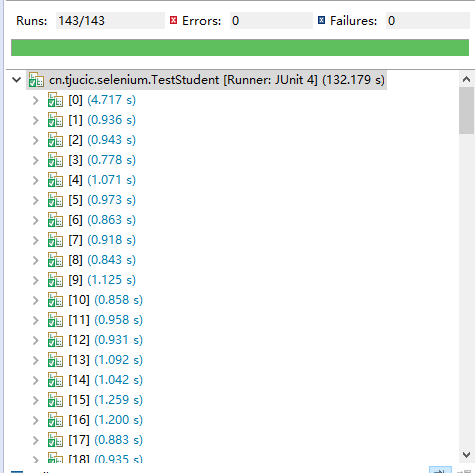
1. 调试分析（在实验过程中遇到的问题以及如何解决）

遇到的问题：第一次测试时，每次运行都会打开100多个geckodriver.exe，导致运行极慢，每次运行都会打开100多个窗口，致使后面的测试用例都会出现TIME\_OUT错误。

解决方法：后来，将driver实例的生成从@Before类中移至@Parameters类中，这样操作之后，每次测试只会生成1个浏览器窗口，运行100多个测试用例，速度极快，很开心。

1. 测试结果（描述输入和输出）

输入为软件测试名单.xlsx，输出如下：



1. 总结

通过这次使用Selenium，了解了web测试的简单流程，加强了对软件测试的性能要求、多参数(Parameters)的理解。