****

**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 刘岳森**

**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）

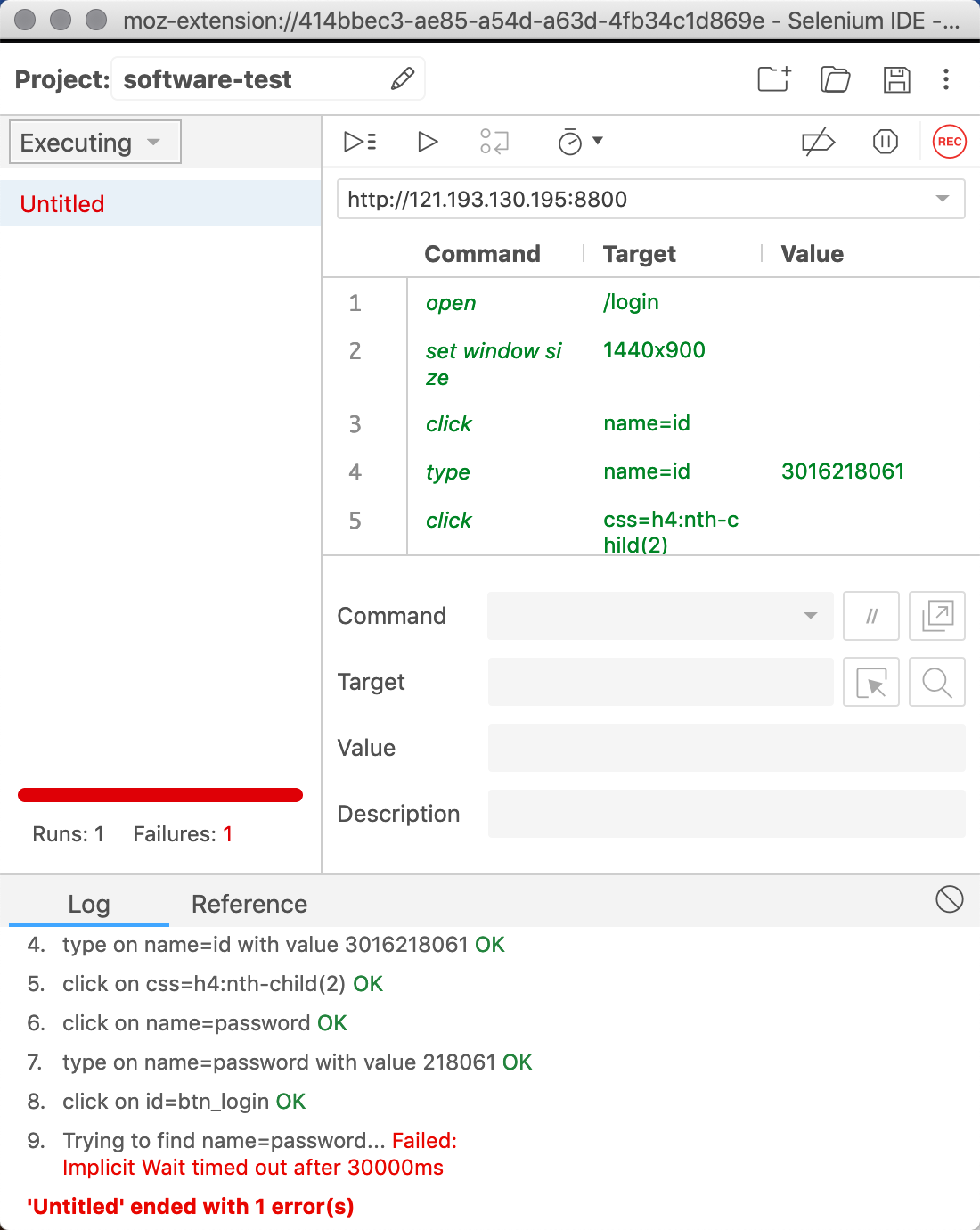
1.安装Selenium, Firefox and SeleniumIDE.

2.记录导出脚本

3.编写java程序用Selenium测试

1. 概要设计（简单描述设计思路，配合UML图）

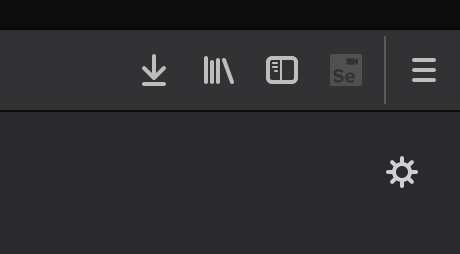
安装Selenium



安装Firefox



安装SeleniumIDE



可以看出来在这个时候我们已经下载好了selenium的插件

1. 详细设计（详细描述具体如何实现，附代码及说明）

**public** **class** Lab2Test {

**private** WebDriver driver;

**private** String baseUrl;

**private** XSSFSheet sheet;

**private** DecimalFormat df = **new** DecimalFormat("0");

**public** Lab2Test() {

}

@Before

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

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

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

**this**.driver = **new** FirefoxDriver();

**this**.baseUrl = "http://121.193.130.195:8800/login";

**this**.driver.manage().timeouts().implicitlyWait(30000L, TimeUnit.***SECONDS***);

XSSFWorkbook workbook = **new** ~~XSSFWorkbook~~("软件测试名单.xlsx");

**this**.sheet = workbook.getSheetAt(0);

}

@Test

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

**this**.driver.get(**this**.baseUrl + "/");

**for**(**int** row\_idx = 2; row\_idx < **this**.sheet.getPhysicalNumberOfRows(); ++row\_idx) {

XSSFRow row = **this**.sheet.getRow(row\_idx);

String username = **this**.df.format(row.getCell(1).getNumericCellValue());

String password = username.substring(username.length() - 6, username.length());

String address = row.getCell(3).toString();

**this**.driver.findElement(By.*name*("id")).sendKeys(**new** CharSequence[]{username});

**this**.driver.findElement(By.*name*("password")).sendKeys(**new** CharSequence[]{password});

**this**.driver.findElement(By.*id*("btn\_login")).sendKeys(**new** CharSequence[]{Keys.***ENTER***});

String student\_git = **this**.driver.findElement(By.*id*("student-git")).getText().toString();

**this**.driver.findElement(By.*id*("btn\_logout")).sendKeys(**new** CharSequence[]{Keys.***ENTER***});

**this**.driver.findElement(By.*id*("btn\_return")).sendKeys(**new** CharSequence[]{Keys.***ENTER***});

~~Assert~~.~~assertEquals~~(address, student\_git);

}

}

}

**public** **class** Lab2 {

**public** **static** **void** main(String[] args){

File excelFile = **new** File("/Users/liuyuesen/Documents/Learning\_materials/大三下/软件测试/实验/实验2/软件测试名单.xlsx");

XSSFWorkbook wb = **null**;

**try** {

wb = **new** XSSFWorkbook(**new** FileInputStream(excelFile));

} **catch** (IOException e) {

e.printStackTrace();

}

**int** numberOfSheets = wb.getNumberOfSheets();

String str = "";

**for** (**int** x = 0; x < numberOfSheets; x++) {

XSSFSheet sheet = wb.getSheetAt(x);

**int** columnNum = 0;

**if** (sheet.getRow(0) != **null**) {

columnNum = sheet.getRow(0).getLastCellNum()

- sheet.getRow(0).getFirstCellNum();

}

**if** (columnNum > 0) {

**for** (Row row : sheet) {

String[] singleRow = **new** String[columnNum];

**int** n = 0;

**for** (**int** i = 0; i < columnNum; i++) {

Cell cell = row.getCell(i, Row.***CREATE\_NULL\_AS\_BLANK***);

**switch** (cell.getCellType()) {

**case** Cell.***CELL\_TYPE\_BLANK***:

singleRow[n] = "";

**if** (cell == **null** || cell.equals("") || cell.getCellType() == HSSFCell.***CELL\_TYPE\_BLANK***) {

System.***out***.print("<Null>|");

} **else** {

System.***out***.print(singleRow[n] + "|");

}

**break**;

**case** Cell.***CELL\_TYPE\_BOOLEAN***:

singleRow[n] = Boolean.*toString*(cell

.getBooleanCellValue());

System.***out***.print(singleRow[n] + "|");

**break**;

**case** Cell.***CELL\_TYPE\_NUMERIC***:

**if** (DateUtil.*isCellDateFormatted*(cell)) {

SimpleDateFormat sdf = **null**;

**if** (cell.getCellStyle().getDataFormat() == HSSFDataFormat

.*getBuiltinFormat*("h:mm")) {

sdf = **new** SimpleDateFormat("HH:mm");

} **else** {

sdf = **new** SimpleDateFormat("yyyy-MM-dd");

}

Date date = cell.getDateCellValue();

System.***out***.print(sdf.format(date) + "|");

} **else** {

cell.setCellType(Cell.***CELL\_TYPE\_STRING***);

String temp = cell.getStringCellValue();

**if** (temp.indexOf(".") > -1) {

singleRow[n] = String.*valueOf*(**new** Double(temp))

.trim();

System.***out***.print(singleRow[n] + "|");

} **else** {

singleRow[n] = temp.trim();

System.***out***.print(singleRow[n] + "|");

}

}

**break**;

**case** Cell.***CELL\_TYPE\_STRING***:

singleRow[n] = cell.getStringCellValue().trim();

System.***out***.print(singleRow[n] + "|");

**break**;

**case** Cell.***CELL\_TYPE\_ERROR***:

singleRow[n] = "";

System.***out***.print(singleRow[n] + "|");

**break**;

**case** Cell.***CELL\_TYPE\_FORMULA***:

cell.setCellType(Cell.***CELL\_TYPE\_STRING***);

String temp = cell.getStringCellValue();

**if** (temp.indexOf(".") > -1) {

temp = String.*valueOf*(**new** Double(temp))

.trim();

Double cny = Double.*parseDouble*(temp);//6.2041

DecimalFormat df = **new** DecimalFormat("0.00");

String CNY = df.format(cny);

System.***out***.print(CNY + "|");

} **else** {

singleRow[n] = temp.trim();

System.***out***.print(singleRow[n] + "|");

}

**default**:

singleRow[n] = "";

**break**;

}

n++;

}

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

}

System.***out***.println("===========================================================Sheet分割线===========================================================");

}

}

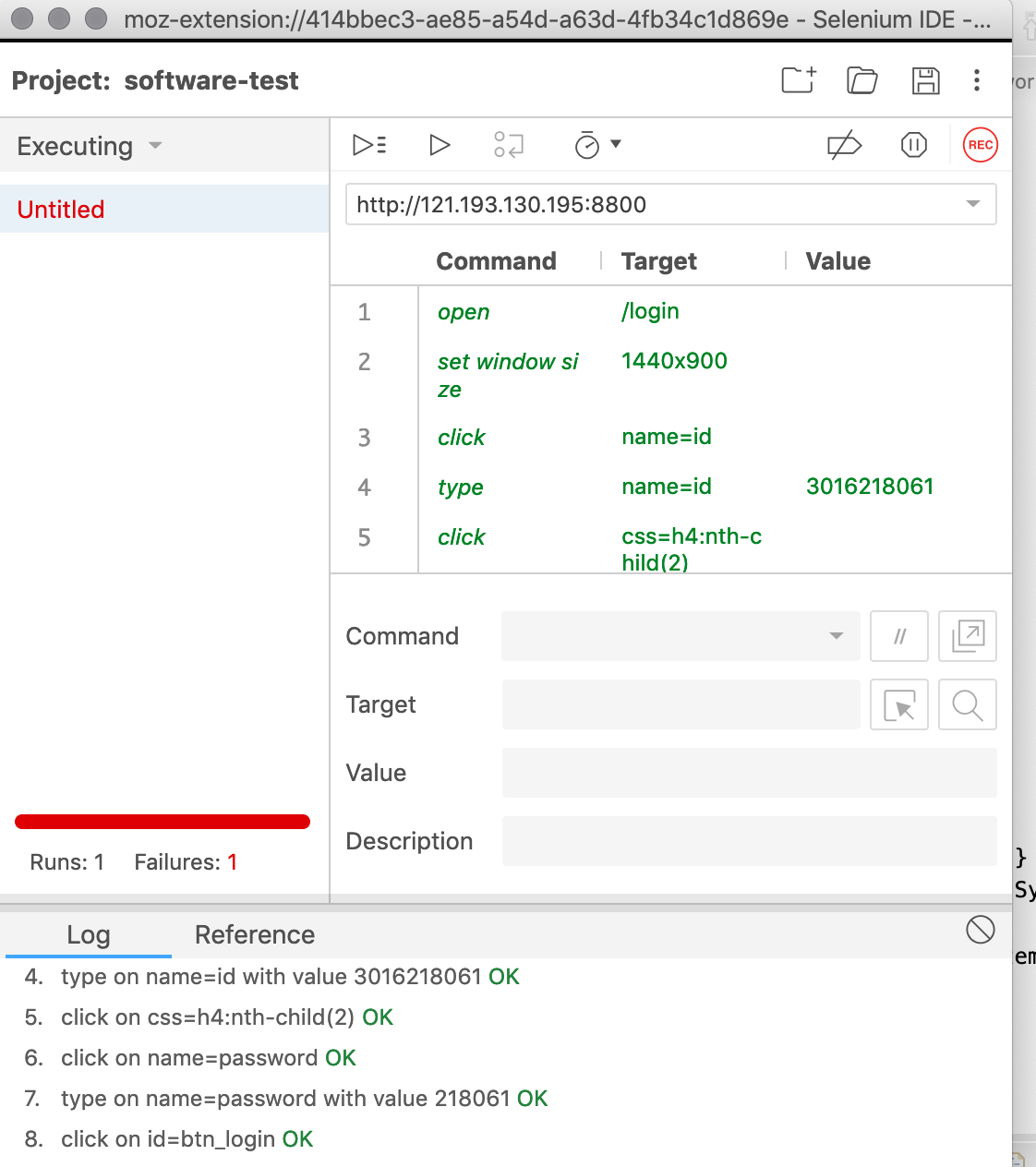
}

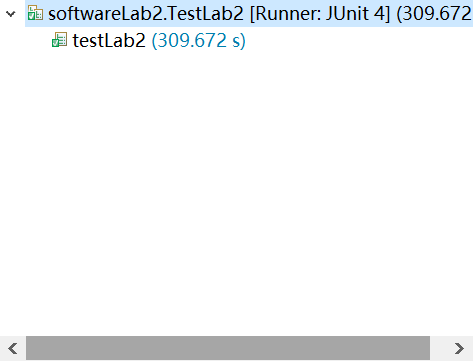
}

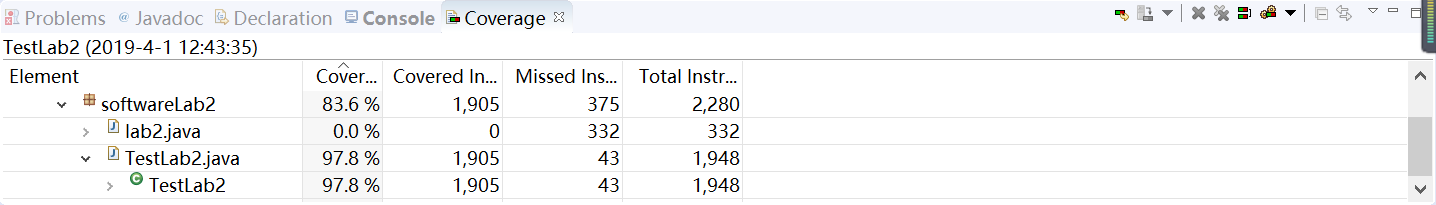
1. 调试分析（在实验过程中遇到的问题以及如何解决）

在Mac当中如果直接执行geckodriver是无法执行的，必须要通过 sudo chmod -x geckodriver来让文件获取执行权限

1. 测试结果（描述输入和输出）







1. 总结

* Selenium的存在简化了我们编写脚本的过程，通过我们对网页的操作就记录下了脚本的执行过程。
* GeckoDriver用于驱动Firefox。文件要执行必须获得执行权限。