|  |
| --- |
| SJTU-SE |
| 测试报告 |
| 第6组 |

|  |
| --- |
| 李 珊 516030910175  王梦瑶 516030910177  陈 诺 516030910199  胡雨奇 516030910257 |

目录

[1. 测试目的 2](#_Toc9776598)

[2. 测试方法 2](#_Toc9776599)

[3. 测试环境 2](#_Toc9776600)

[4. 测试前准备 2](#_Toc9776601)

[1. Selenium使用 2](#_Toc9776602)

[2. 测试代码架构 2](#_Toc9776603)

[3. 登录逻辑 2](#_Toc9776604)

[5. 测试内容 3](#_Toc9776605)

[5.1 页面跳转 3](#_Toc9776606)

[5.2 按钮 3](#_Toc9776607)

[5.3 输入框 3](#_Toc9776608)

[5.4 多选框 3](#_Toc9776609)

[5.5 下拉框 3](#_Toc9776610)

[5.6 文件上传 3](#_Toc9776611)

[5.6.1 上传图片 3](#_Toc9776612)

[5.6.2 测试逻辑 3](#_Toc9776613)

[5.6.3 测试结果 4](#_Toc9776614)

[4.7 模态框 5](#_Toc9776615)

[5. 测试结果及总结 5](#_Toc9776616)

[6. 参考 6](#_Toc9776617)

# 1. 测试目的

1. 练习和掌握web GUI测试的一般步骤
2. 掌握使用Intellij和Selenium进行测试的方法
3. 通过测试验证目标网站的正确性

# 2. 测试方法

使用Intellij和Selenium来进行web GUI测试。

通过针对网站代码编写测试代码，来验证网站基础功能的正确性。

# 3. 测试环境

1. 系统：Windows10-64bit
2. IDE： Intellij
3. 库：Selenium 3.141.59 Maven

# 4. 测试前准备

## 1. Selenium使用

## 2. 测试代码架构

## 3. 登录逻辑

public class LoginUtil {

void addCookie(ChromeDriver driver);

void getCookie(ChromeDriver driver);

}

首次登录时使用addCookie：跳转入登录页面自动填写用户名密码登录，登录完成后，使用driver.manage().getCookies()获取cookie并写入本地文件douban.cookie.txt中。

再次登录时使用getCookie：读取本地文件douban.cookie.txt中的字段生成cookie，使用driver.manage().addCookie(cookie)使cookie生效完成自动登录。

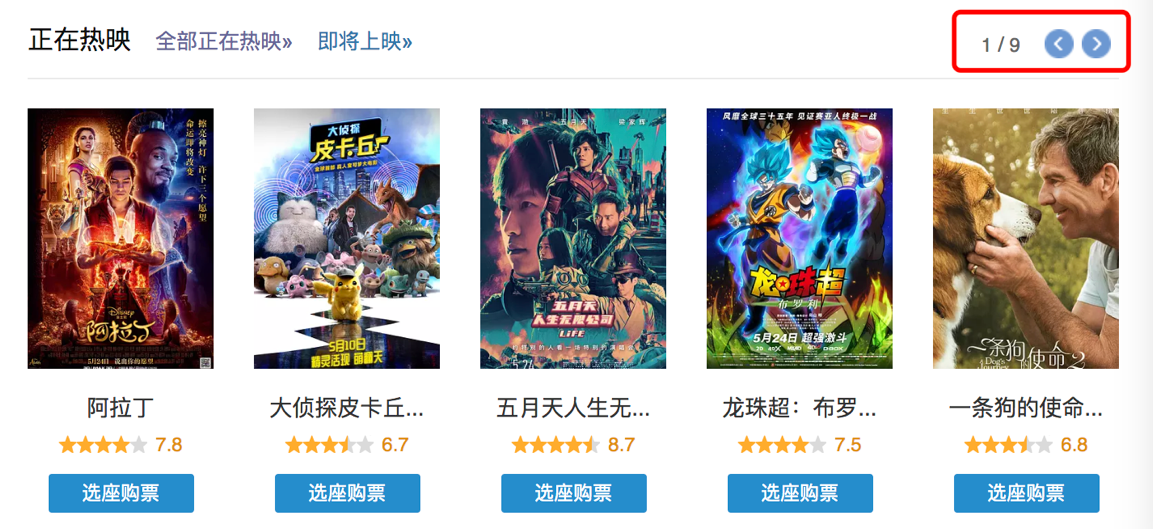
# 5. 测试内容

## 5.1 页面跳转

**5.2 按钮** ButtonTest.java

**5.2.1 [正在热映]组件**

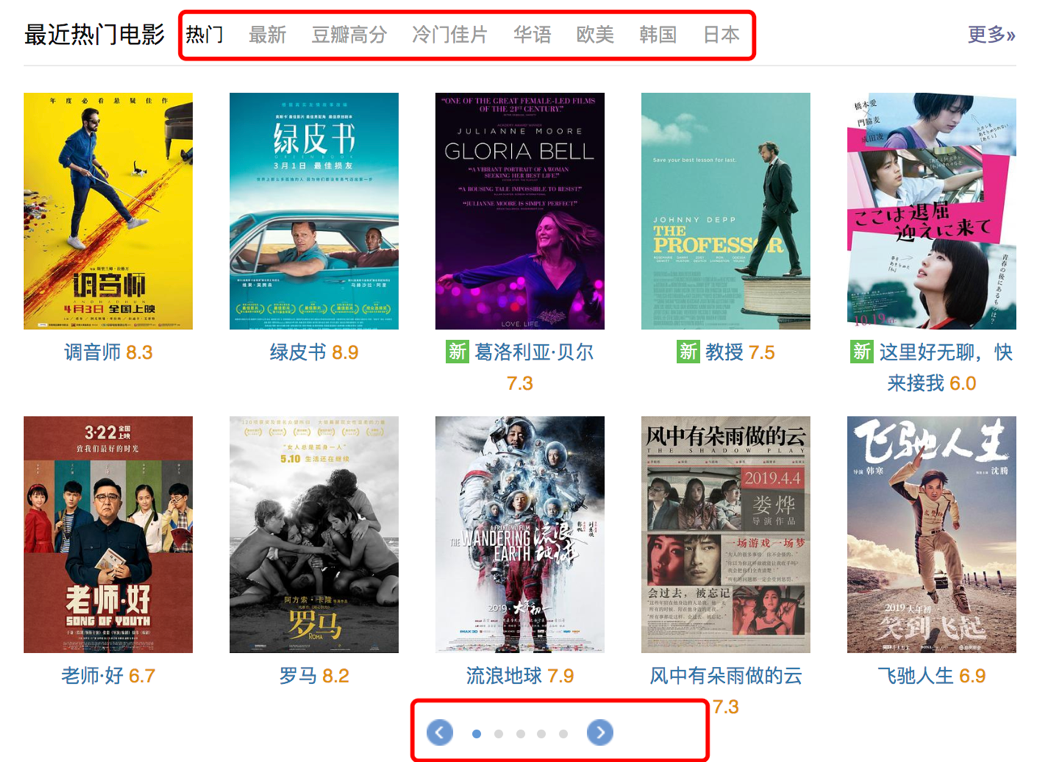
testSlideButton1() 测试翻页按钮，判断页数以及窗口中影片内容是否改变



**5.2.2 [最近热门电影]组件**

testSlideButton2("movie") 测试翻页按钮，判断idx以及窗口中影片内容是否改变

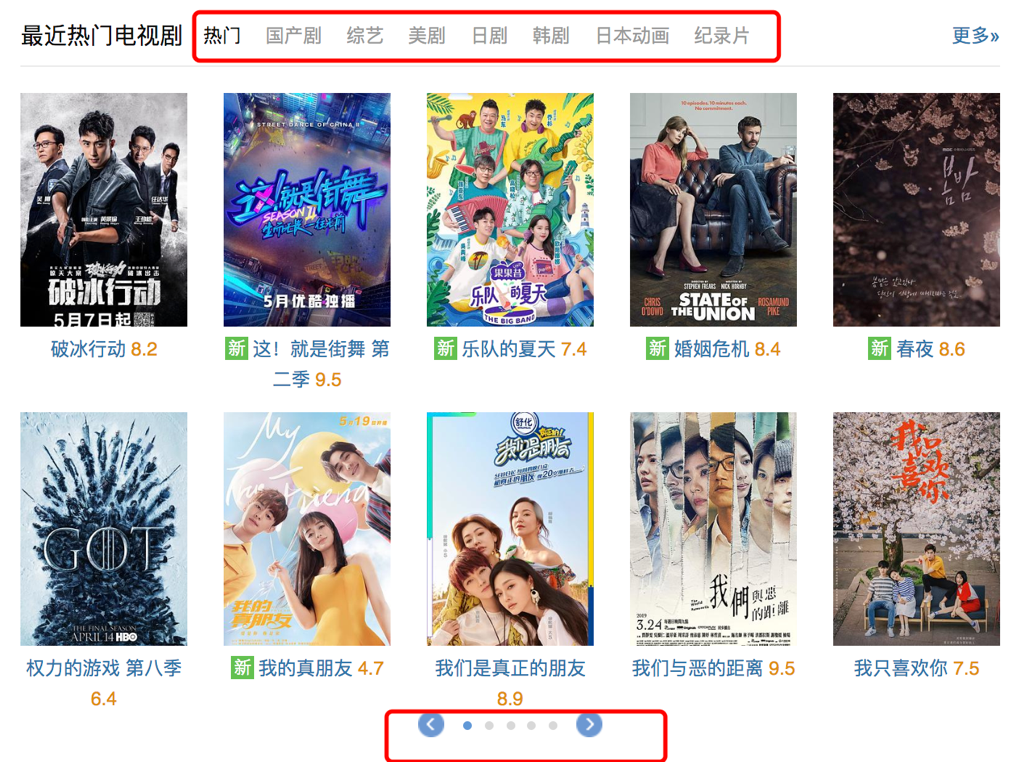
testTagButton("movie") 测试标签按钮，判断窗口中影片内容是否改变



**5.2.3 [最近热门电视剧]组件** （组件结构同热门电影）

testSlideButton2("tv")

testTagButton("tv")



**5.2.4 [热门推荐]组件**

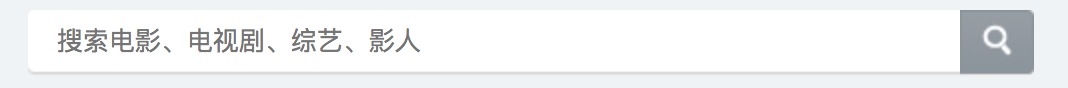
testSlideButton3() 测试定时翻页及测试翻页按钮，判断页数以及窗口中影片内容是否改变



**5.3 输入框** InputTest.java

**5.3.1 搜索组件**

testSearch() 填写输入框并点击搜索按钮，检查搜索结果并测试翻页按钮







**5.3.2 评论组件**

testComment("movie")

测试浮窗并点击按钮[想看]

Actions action = new Actions(driver);

action.moveToElement(item).perform();

WebElement btn=driver.findElementByXPath(String.format("//div[@class='gaia gaia-lite gaia-%s slide-mode']/div[1]/div[1]/div[1]/p[3]/a[1]",topic));

btn.click();



填写评论，并点击仅自己可见后按保存提交

WebElement inputBox=driver.findElementByXPath("//textarea[@id='comment']");

inputBox.sendKeys("233");

WebElement checkBox=driver.findElementByXPath("//input[@id='inp-private']");

checkBox.click();

WebElement submitBtn=driver.findElementByXPath("//div[@id='submits']/span/input");

submitBtn.click();



为了能够重复测试，删除该条评论：点击进入该电影的详情页，点击删除，并操作alert窗口确认删除

driver.navigate().to(movieLink);

WebElement cancelBtn=

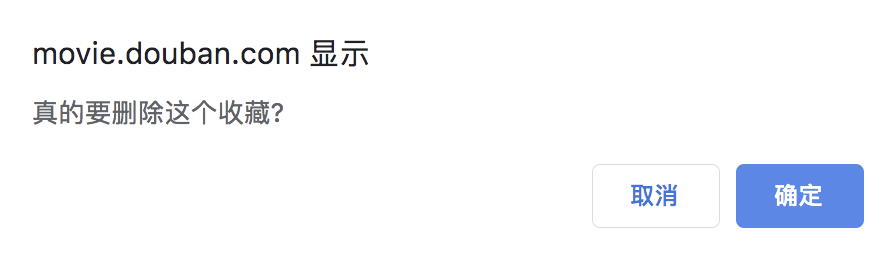
driver.findElementByXPath("//div[@id='interest\_sect\_level']/div/span/span[2]/form/input");

cancelBtn.click();

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

alert.accept();





## 5.4 多选框

测试文件:CheckBoxTest.java

**5.4.1 [分类]组件**

testCheckBox1() 测试可播放和我没看过的多选框，看该组件状态是否可以被选中

找到多选框的列表

List<WebElement> checkBoxInput = driver.findElementsByCssSelector(".check label.checkbox span");

WebElement checkbox1 = checkBoxInput.get(0);



点击查看css变化

assert (checkbox1.getAttribute("class").equals("checkbox\_\_input"));

checkbox1.click();

assert (checkbox1.getAttribute("class").equals("checkbox\_\_input is-checked is-focus"));

checkbox1.click();

assert (checkbox1.getAttribute("class").equals("checkbox\_\_input is-checked"));



**5.4.2 [选电影]组件**

testCheckBox2() 测试可播放和我没看过的多选框，看该组件状态是否被选中,看该多选框是否可以同时被多选

点击一个：

WebElement e2=driver.findElementByName("watched");

assert !e2.isSelected();

e2.click();

assert e2.isSelected();

e2.click();

assert !e2.isSelected();

点击两个：

e2.click();e3.click();

assert (e2.isSelected()&&e3.isSelected());



两个同时不选：

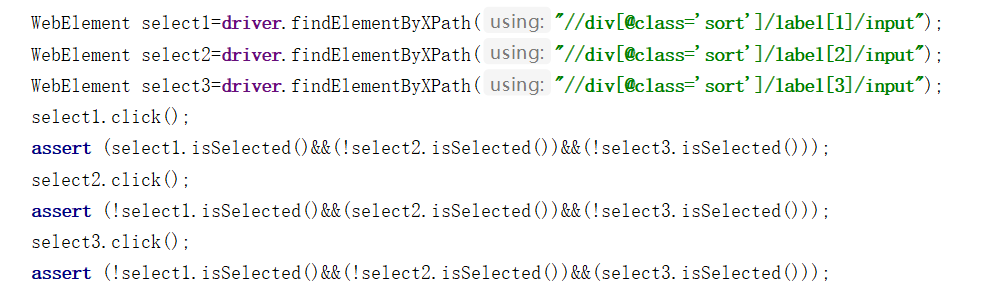
e2.click();e3.click();

assert ((!e2.isSelected())&&(!e3.isSelected()));

**5.2.3 [电视剧]组件**

testCheckBox3()测试前面的单选框，可播放和我没看过的多选框，看前三个单选框是否每次只能选中一个，看多选框是否可以同时被多选

单选框测试：



检查是否选中一个会自动取消其他两个的选中



## 5.5 下拉框

测试文件:DropBoxTest.java

**5.5.1 下拉框组件**

testDropBox() 点击下拉框并点击评分按钮，检查点击结果



在没有点击这个按钮的时候，检查这个组件的属性，并查找下拉框的组件

assert (d1.getAttribute("class").equals("angle-down"));

assert !doesElementExist("//div[@class='range-dropdown']");

是不存在的

点击之后是状态改变并且下拉框组件出现

d1.click();

assert (d1.getAttribute("class").equals("angle-up"));

assert doesElementExist("//div[@class='range-dropdown']");



拖动评分区间，值会改变

WebElement r3=driver.findElementByXPath("//div[@class='slider-wrap']/div[3]");

r3.click();

assert (d2.getAttribute("value").equals("2,10"));



## 5.6 文件上传

测试文件：.\src\main\java\test\UploadFileTest.java

### 5.6.1 上传图片

网页位置：<https://accounts.douban.com/passport/setting>

测试内容：上传头像

### 5.6.2 测试逻辑

进入网页<https://accounts.douban.com/passport/setting>后，选择上传头像，并给弹出的文件夹输入目标图片位置，在浮动窗口中选择确定完成文件上传。



### 5.6.3 测试结果

通过在Intellij中以debug模式运行并打断点查看测试结果。

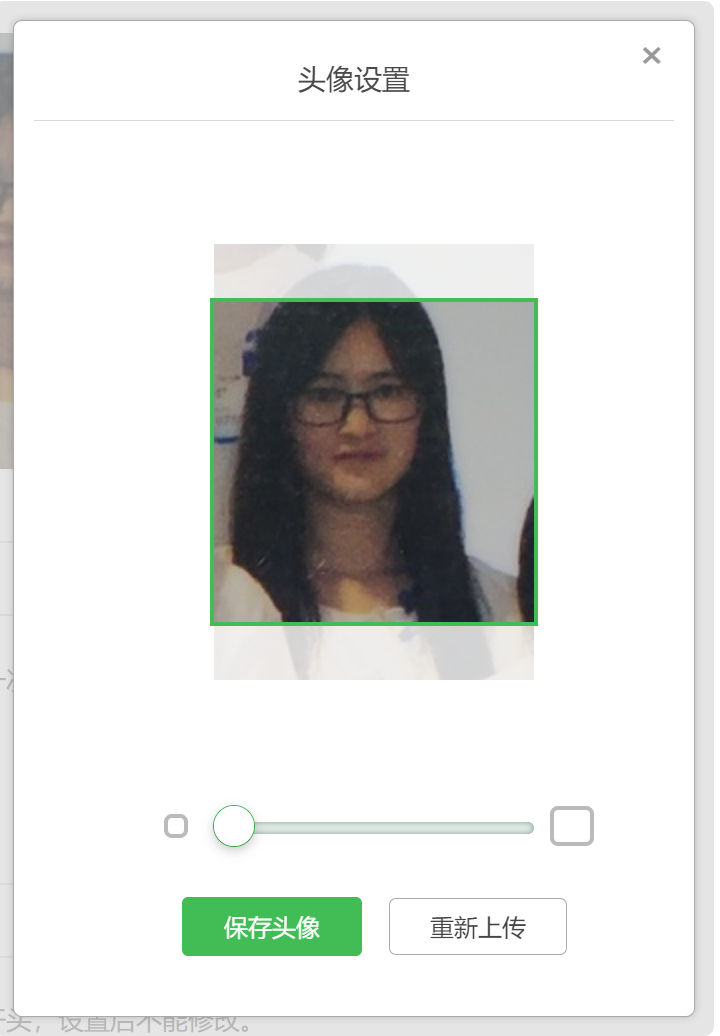


1. 进入个人设置页



2. 给弹出的文件夹输入图片路径

3. 在浮窗中确认图片



4. 测试结束

## 4.7 模态框

# 5. 测试结果及总结

# 6. 参考