

UI自动化测试框架之Selenium关键字驱动

使用工具：eclipse

用到的第三方jar包：poi.jar(操作Excel);selenium.jar

理解难点：java反射机制

一、框架构思

1、编写脚本

首先我们来写一个登陆开源中国的脚本

```
public class Login_Script {
    public static WebDriver driver=null;

    public static void main(String []args) throws InterruptedException{
//        启动火狐浏览器
        driver= new FirefoxDriver();
//        最大化
        driver.manage().window().maximize();
//        打开开源中国网址
        driver.get("http://www.oschina.net/");
//        点击登录
        driver.findElement(By.xpath("//*[@id='OSC_Userbar']/a[1]")).click();
//        输入用户名
        driver.findElement(By.xpath("//*[@id='f_email']")).sendKeys("XXXXXXB");
//        输入密码
        driver.findElement(By.xpath("//*[@id='f_pwd']")).sendKeys("XXXXXXXA");
//        点击登录按钮
        driver.findElement(By.xpath("//*[@id='login_osc']/table/tbody/tr[7]/td/input")).click();
//        Thread.sleep(30);
//        点击退出按钮
        driver.findElement(By.xpath("//*[@id='OSC_Userbar']/a[3]")).click();
//        关闭浏览器
        driver.quit();
    }
}
```

2、脚本分析

这是登陆的场景

操作步骤

第一步：启动浏览器

第二步：输入网址

第四步：点击登录

第五步：输入用户名

第六步：输入密码

第七步：点击登录按钮

第八步：点击退出

第九步：关闭浏览器

3、使用excel

建立一个excel

CaseSuite	CaseId	Description	KeyWords
login	id_001	打开浏览器	OpenBrowser
login	id_002	输入网址	Navigate
login	id_003	点击登录	Login_Click
login	id_004	输入用户名	Input_Name
login	id_005	输入密码	Input_Password
login	id_006	点击登录按钮	Login_Button
login	id_007	点击退出	Logout_Click
login	id_008	关闭浏览器	CloseBrowser

在java中创建一个操作Excel的类，主要实现是对Excel的读和写，主要代码如下：

```
public class ExcelUtils {

    public static HSSFWorkbook ExcelSheet;
    public static HSSFWorkbook ExcelBook;
    public static HSSFRow Row;
```

```

public static HSSFCell Cell;

public static void setExcelFile(String Path,String SheetName) throws Exception{
    FileInputStream ExcelFile=new FileInputStream(Path);
    ExcelBook=new HSSFWorkbook(ExcelFile);
    ExcelSheet=ExcelBook.getSheet(SheetName);
}

public static void setCellData(String Result, int RowNum, int ColNum,String Path) throws Exception{
    Row = ExcelSheet.getRow(RowNum);
    Cell = Row.getCell(ColNum, Row.RETURN_BLANK_AS_NULL);
    if (Cell == null) {
        Cell = Row.createCell(ColNum);
        Cell.setCellValue(Result);
    } else {
        Cell.setCellValue(Result);
    }
    FileOutputStream fileOut = new FileOutputStream(Path);
    ExcelBook.write(fileOut);
    fileOut.flush();
    fileOut.close();
}

public static String getCellDate(int RowNum,int CloNum){
    Cell=ExcelSheet.getRow(RowNum).getCell(CloNum);
    String cellData=Cell.getStringCellValue();
    return cellData;
}
}

```

5、新建一个ActionKeyWords类

```

public class ActionKeyWords {
    public static WebDriver driver=null;

// 启动浏览器并最大化
    public static void OpenBrowser (){
        driver= new FirefoxDriver();
        driver.manage().window().maximize();
    }

// 打开开源中国网址
    public static void Navigate (){
        driver.get("http://www.oschina.net/");
    }

// 点击登录
    public static void Login_Click (){
        driver.findElement(By.xpath("//*[@id='OSC_Userbar']/a[1]")).click();
    }

// 输入用户名
    public static void Input_Name (){
        driver.findElement(By.xpath("//*[@id='f_email']")).sendKeys("lcmamburensheng@126.com");
    }

// 输入密码
    public static void Input_Password (){
        driver.findElement(By.xpath("//*[@id='f_pwd']")).sendKeys("12307575lc");
    }

// 点击登录按钮
    public static void Login_Button (){
        driver.findElement(By.xpath("//*[@id='login_osc']/table/tbody/tr[7]/td/input")).click();
    }

// 点击退出按钮
    public static void Logout_Click (){

```

```

        driver.findElement(By.xpath("//*[@id='OSC_Userbar']/a[3]")).click();
    }

// 关闭浏览器
public static void CloseBrowser (){
    driver.quit();
}

}
5、修改Login_Script脚本

public class Login_Script {

    public static void main(String []args) throws Exception{

        ExcelUtils.setExcelFile("D:\\data\\TestData.xls", "steps");
        ActionKeyWords actionKeyWords= new ActionKeyWords();
        String Keywords=null;
        for(int RowNum=1;RowNum<=ExcelUtils.getLastRowNums();RowNum++){
            Keywords=ExcelUtils.getCellDate(RowNum, 3);
            if(Keywords.trim().equals("OpenBrowser")){
                actionKeyWords.OpenBrowser();
            }else if(Keywords.trim().equals("Navigate")){
                actionKeyWords.Navigate();
            }else if(Keywords.trim().equals("Login_Click")){
                actionKeyWords.Login_Click();
            }else if(Keywords.trim().equals("Input_Name")){
                actionKeyWords.Input_Name();
            }else if(Keywords.trim().equals("Input_Password")){
                actionKeyWords.Input_Password();
            }else if(Keywords.trim().equals("Login_Button")){
                actionKeyWords.Login_Button();
            }else if(Keywords.trim().equals("Logout_Click")){
                actionKeyWords.Logout_Click();
            }else if(Keywords.trim().equals("CloseBrowser")){
                actionKeyWords.CloseBrowser();
            }
        }
    }
}
}

```

这样代码的框架就基本已经搭建起来了，代码结构如下：

```

└─ KeyDrive
   └─ src
      ├── config
      │   └─ ActionKeyWords.java
      ├── script
      │   └─ Login_Script.java
      └─ utility
          └─ ExcelUtils.java

```

二、程序代码优化

1、优化Login_Script 类中的代码

这里用到了反射机制

```

public class Login_Script {

    public static ActionKeyWords actionKeyWords;
    public static String Keywords=null;
    public static Method[] method;
    public Login_Script(){
        actionKeyWords= new ActionKeyWords();
        method=actionKeyWords.getClass().getMethods();
    }

    public static void main(String []args) throws Exception{
        ExcelUtils.setExcelFile("D:\\data\\TestData.xls", "steps");
    }
}

```

```

        new Login_Script();
        for(int rowNum=1;RowNum<=ExcelUtils.getLastRowNums();RowNum++){
            Keywords=ExcelUtils.getCellDate(RowNum, 3);

            login_action();
        }
    }

    public static void login_action(){
        for(int i=0;i<method.length;i++){
//            System.out.println(method[i].getName()+" "+actionKeyWords+Keywords);
            if(method[i].getName().trim().equals(Keywords)){
                try {
                    method[i].invoke(actionKeyWords);

                } catch (IllegalAccessException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
                } catch (IllegalArgumentException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
                } catch (InvocationTargetException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
                }
            }
        }
    }
}

```

2、将程序中的常量统一管理

例如：网页的地址，账户、密码，excel路径，这里我们在文件下面建立一个

```

public class Contants {

    public static String url="http://www.oschina.net/";
    public static String excelFile="D:\\data\\";
    public static String excelName="TestData.xls";
    public static String excelSheet="steps";
    public static int excelKWCloNum=3;
    public static String userName="XXXXXXA";
    public static String userPassword="XXXXXB";

}

```

3、增加对象库

下面我们看一下ActionKeyWords类中定位元素的路径 是在代码里面的，如果每次去修改的定位路径的是时候都要修改代码，为了便于维护，我们将这些元素的对象放在一个文件中,同时我们在Excel增加一列 Page Objects，这样程序根据Excel中的Page Objects，去文件中读取相应的元素，这里我们增加一个类OrpUtil,读取元素的对象

Home Page Objects

Userbar_login=//*[@id='OSC_Userbar']/a[1]

Userbar_logout=//div[@id='OSC_Userbar']/a[3]

#Login Page Objects

Input_name=//*[@id='f_email']

Input_password=//*[@id='f_pwd']

Login_button=//*[@id='login_osc']/table/tbody/tr[7]/td/input

//OrpUtil类

```

public class OrpUtil {

    public static String readValue(String a){
        Properties pro=new Properties();
        String popath=Contants.ObjectReUrl;
        String value=null;
        try {
            InputStream in =new BufferedInputStream(new FileInputStream(popath));
            pro.load(in);

```

```

        value=pro.getProperty(a);

    } catch (FileNotFoundException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    return value;
}
}

```

优化后的ActionKeyWords类

```

public class ActionKeyWords {
    public static WebDriver driver=null;

    // 启动浏览器并最大化
    public static void OpenBrowser (String OR){
        System.setProperty("webdriver.chrome.driver", ".//server//chromedriver.exe");
        driver= new ChromeDriver();
        driver.manage().window().maximize();

    }

    // 打开开源中国网址
    public static void Navigate (String OR){
        driver.get(Contants.url);
    }

    // 点击登录
    public static void Login_Click (String OR){
        driver.findElement(By.xpath(OrpUtil.readValue(OR))).click();
    }

    // 输入用户名
    public static void Input_Name (String OR){
        driver.findElement(By.xpath(OrpUtil.readValue(OR))).clear();
        driver.findElement(By.xpath(OrpUtil.readValue(OR))).sendKeys(Contants.userName);
    }

    // 输入密码
    public static void Input_Password (String OR){
        driver.findElement(By.xpath(OrpUtil.readValue(OR))).click();
        driver.findElement(By.xpath(OrpUtil.readValue(OR))).sendKeys(Contants.userPassword);
    }

    // 点击登录按钮
    public static void Login_Button (String OR){
        driver.findElement(By.xpath(OrpUtil.readValue(OR))).click();
    }

    // 点击退出按钮
    public static void Logout_Click (String OR){
        try {
            Thread.sleep(300);
            driver.findElement(By.xpath(OrpUtil.readValue(OR))).click();
        } catch (InterruptedException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }

    }

    // 关闭浏览器
    public static void CloseBrowser (String OR){

```

```

    driver.quit();
}

}

```

这个OR的值是从Excel中读取的

TestCaseId	Ts_id	Description	PageObject	Action_Keyword
login_001	Ts_001	打开浏览器		OpenBrowser
login_001	Ts_002	导航栏输入地址		Navigate
login_001	Ts_003	点击登录	Userbar_login	Login_Click
login_001	Ts_004	输入用户名	Input_name	Input_Name
login_001	Ts_005	输入密码	Input_password	Input_Password
login_001	Ts_006	点击登录按钮	Login_button	Login_Button
login_001	Ts_007	退出	Userbar_logout	Logout_Click
login_001	Ts_008	关闭浏览器		CloseBrowser

4、增加测试场景

从Excel中我们可以看到，这操作是对应的用例编写中的我们的操作步骤，在用例设计的时候还有测试场景和结果，这里我们先增加个场景在EXCEL中增加一个名称为Suite的Sheet页

TestSuiteId	Description	Runmode
login_001	登陆场景测试	YES
login_002	不运行	YES
login_003	不运行	NO
login_004	不运行	NO
login_005	不运行	NO

我们程序的运行逻辑是循环读取Suite页中的Runmode,当为YES时根据对应的TestSuiteID去读取对应的Steps页中的操作在步骤，进行运行，程序核心逻辑代码如下：

```

public static void main(String []args) throws Exception{
    ExcelUtils.setExcelFile(Contants.excelFile+Contants.excelName );
    new Login_Script();
    for(int j=1;j<=ExcelUtils.getLastRowNums(Contants.suitSheet);j++){
        String Runmode=ExcelUtils.getCellDate(j, Contants.suitRunmode,Contants.suitSheet);
        String suitTestSuiteId=ExcelUtils.getCellDate(j, Contants.suitTestSuiteId,Contants.suitSheet);
        if(Runmode.equals("YES")){
            for(int sRowNum=1;sRowNum<=ExcelUtils.getLastRowNums(Contants.caseSheet);sRowNum++){
                String stepTestSuiteId=ExcelUtils.getCellDate(sRowNum, Contants.stepTestSuiteId,Contants.caseSheet);
                System.out.println(ExcelUtils.getCellDate(sRowNum, Contants.excelKWCloNum,Contants.caseSheet));
                if(stepTestSuiteId.trim().equals(suitTestSuiteId)){
                    Keywords=ExcelUtils.getCellDate(sRowNum, Contants.excelKWCloNum,Contants.caseSheet);
                    r=ExcelUtils.getCellDate(sRowNum, Contants.excelPOCloNum,Contants.caseSheet);
                    login_action();
                }
            }
        }
        else{
            System.out.println("没有要执行的用例");
            continue;
        }
    }
}

```

5、增加测试结果

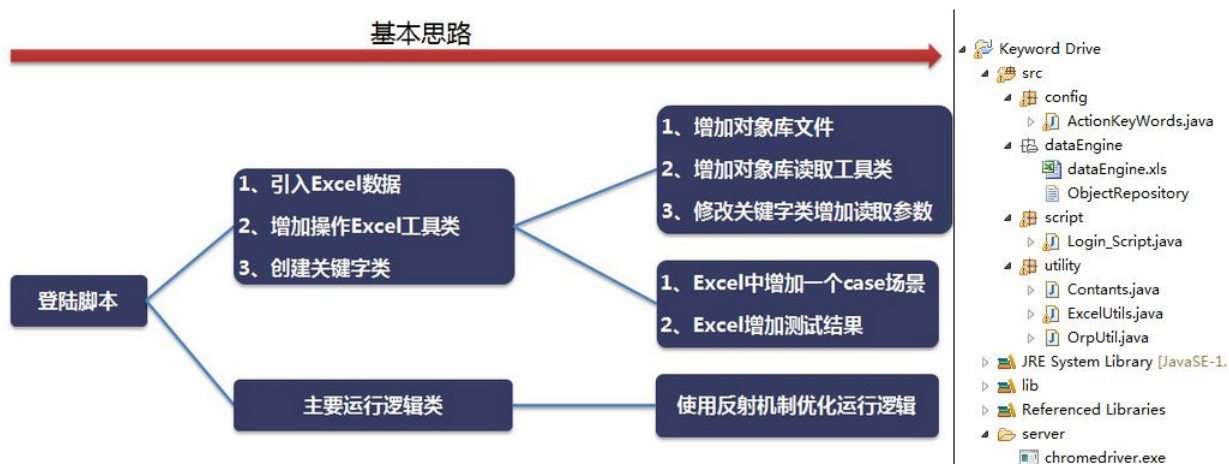
在Excel中新增一列Resut

TestSuiteId	Ts_id	Description	PageObject	Action_Keyword	Results
login_001	Ts_001	打开浏览器		OpenBrowser	PASS
login_001	Ts_002	导航栏输入地址		Navigate	PASS
login_001	Ts_003	点击登录	Userbar_login	Login_Click	PASS
login_001	Ts_004	输入用户名	Input_name	Input_Name	PASS
login_001	Ts_005	输入密码	Input_password	Input_Password	PASS
login_001	Ts_006	点击登录按钮	Login_button	Login_Button	PASS
login_001	Ts_007	退出	Userbar_logout	Logout_Click	FAIL
login_001	Ts_008	关闭浏览器		CloseBrowser	FAIL

TestSuiteId	Description	Runmode	Results
login_001	登陆场景测试	YES	FAIL
login_002	不运行	NO	
login_003	不运行	NO	
login_004	不运行	NO	
login_005	不运行	NO	

在Login_Script中定义一个boolean类型的变量bResult，默认是true在各个地方try,,catch,当出现异常的时候在bResult赋值为false,在Excel工具类中增加一个写入excel值得方法

这样我们的的框架就初步搭建好了，下面我们回归一下基本思路



代码地址：<http://git.oschina.net/hellotester/SeleniumKeywordDrive>

参考文献：<http://www.toolsqa.com/selenium-webdriver/keyword-driven-framework/introduction/>