selenium webdriver学习

作者: qi_ling2005 http://jarvi.iteye.com

We are Evil Testers!

目 录

1. Selenium-webdriver

1.1 selenium webdriver学习（一）----------快速开始

发表时间: 2012-03-09 关键字: selenium webdriver 学习

selenium webdriver学习历程（一）----------快速开始

学习selenium已经两年了，从1.X到2.X，一直在关注它。中间由于工作原因中断了一段时间，但是一直无法割舍，最近又去官网看了一下，更新还挺快的。selenium1.X的时代将被取代，selenium-webdriver的大航海时代开始了。。。

安装selenium webdriver（eclipse+jdk+selenium webdriver2.20+firefox 10）

1、安装firefox，本人使用firefox10。确保firefox安装在默认环境下（不是的话会报错）。

2、安装jdk，确保安装了jdk，本人喜欢使用java。但selenium webdriver也支持其它语言，如ruby、python、C#等。

3、安装eclipse，个人喜好。

4、安装selenium webdriver。解压下载的selenium webdriver包，可以在eclipse建一个user library，便与项目的引入。

第一个test

现在以第一个selenium webdriver的test来感受一下它的魅力。

```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;



public class FirstExampe {
```

```
        public static void main(String[] args) {
                WebDriver driver = new FirefoxDriver();



        driver.get("http://www.google.com.hk");
        WebElement element = driver.findElement(By.name("q"));
        element.sendKeys("hello Selenium!");
        element.submit();
        try {
                        Thread.sleep(3000);
                } catch (InterruptedException e) {
                        e.printStackTrace();
                }
        System.out.println("Page title is: " + driver.getTitle());

        driver.quit();
        }

}
```

正常运行后，这几行代码将会打开firefox浏览器，然后转跳到google首页。在搜索框中输入hello Selenium并提交搜索结果。等待3秒后会在命令行打印出当前页面的title，输出如下：

```
Page title is: hello Selenium! - Google 搜尋
```

并关闭ff浏览器。

1.2 selenium webdriver学习（二）————对浏览器的简单操作

发表时间: 2012-03-09

selenium webdriver对浏览器的简单操作

打开一个测试浏览器

对浏览器进行操作首先需要打开一个浏览器，接下来才能对浏览器进行操作。但要注意的是，因为Chrome Driver是Chromium 项目自己支持和维护的，所以你必需另外下载安装Chrome Driver，详细介绍查下他们的wiki 。

```
import java.io.File;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxBinary;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.ie.InternetExplorerDriver;

public class OpenBrowsers {


        public static void main(String[] args) {
                //打开默认路径的firefox
                WebDriver diver = new FirefoxDriver();

                //打开指定路径的firefox,方法1
                System.setProperty("webdriver.firefox.bin","D:\\Program Files\\Mozilla Firefox\
                WebDriver dr = new FirefoxDriver();

                //打开指定路径的firefox,方法2
                File pathToFirefoxBinary = new File("D:\\Program Files\\Mozilla Firefox\\firefo
                FirefoxBinary firefoxbin = new FirefoxBinary(pathToFirefoxBinary);
                WebDriver driver1 = new FirefoxDriver(firefoxbin,null);
```

```
                //打开ie
                WebDriver ie_driver = new InternetExplorerDriver();

                //打开chrome
                System.setProperty("webdriver.chrome.driver", "D:\\chromedriver.exe");
                System.setProperty("webdriver.chrome.bin",
                                        "C:\\Documents and Settings\\gongjf\\Local Setting
                                        +"\\Application Data\\Google\\Chrome\\Application\


        }

}
```

打开指定路经ie和chrome方法和ff一样。

打开1个具体的url

打开一个浏览器后，我们需要跳转到特定的url下，看下面代码：

```
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;

public class OpenUrl {
        public static void main(String []args){
                String url = "http://www.51.com";
                WebDriver driver = new FirefoxDriver();

                //用get方法
                driver.get(url);

                //用navigate方法，然后再调用to方法
                driver.navigate().to(url);
        }
}
```

如何关闭浏览器

测试完成后，需要关闭浏览器

```
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;

public class CloseBrowser {
        public static void main(String []args){
                String url = "http://www.51.com";
                WebDriver driver = new FirefoxDriver();

                driver.get(url);

                //用quit方法
                driver.quit();

                //用close方法
                driver.close();
                }
}
```

如何返回当前页面的url和title

有时候我们需要返回当前页面的url或者title做一些验证性的操作等。代码如下：

```
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;

public class GetUrlAndTitle {
        public static void main(String []args){
                String url = "http://www.51.com";
                WebDriver driver = new FirefoxDriver();
```

```
                driver.get(url);

                //得到title
                String title = driver.getTitle();

                //得到当前页面url
                String currentUrl = driver.getCurrentUrl();

                //输出title和currenturl
                System.out.println(title+"\n"+currentUrl);

                }
}
```

其他方法

- getWindowHandle()　返回当前的浏览器的窗口句柄
- getWindowHandles()　返回当前的浏览器的所有窗口句柄
- getPageSource()　　返回当前页面的源码

小结

从上面代码可以看出操作浏览器的主要方法都来自org.openqa.selenium.WebDriver这个接口中。看了一下源代码这些方法都是在org.openqa.selenium.remote.RemoteWebDriver这个类中实现的，然后不同浏览的driver类继承RemoteWebDriver。

1.3 selenium webdriver学习（三）----------执行js脚本

发表时间: 2012-03-09

在用selenium 1.X的时候常常会用到getEval()方法来执行一段js脚本来对页面进行处理，以处理一些遇到的问题。当然selenium webdriver也提供这样的一个方法:executeScript()

```
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.WebDriver;

public class  SimpleExample {


        public static void main(String[] args) {
                WebDriver driver = new FirefoxDriver();

              ((JavascriptExecutor)driver).executeScript("alert(\"hello,this is a alert!\")");
        }

}
```

上面是一个最简单的例子，打开一个浏览器，然后弹层一个alert框。注意这里的driver要被强制转换成JavascriptExecutor。

下面演示在打开51.com首页如何得到帐号输入框中显示的字符，并打印输出。

```
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.WebDriver;

public class FirstExampe {


        public static void main(String[] args) {
                WebDriver driver = new FirefoxDriver();
                driver.get("http://www.51.com");
                String js = "var user_input = document.getElementById(\"passport_51_user\").tit
```

```
                String title = (String)((JavascriptExecutor)driver).executeScript( js);
                System.out.println(title);
        }

}
```

输出结果为：

```
用户名/彩虹号/邮箱
```

1.4 selenium webdriver学习（四）----------定位页面元素

发表时间: 2012-03-10 关键字: selenium, webdriver, 定位页面元素, findElement, By

selenium-webdriver提供了强大的元素定位方法，支持以下三种方法。

- 单个对象的定位方法
- 多个对象的定位方法
- 层级定位

定位单个元素

在定位单个元素时,selenium-webdriver提示了如下一些方法对元素进行定位。

- By.className(className))
- By.cssSelector(selector)
- By.id(id)
- By.linkText(linkText)
- By.name(name)
- By.partialLinkText(linkText)
- By.tagName(name)
- By.xpath(xpathExpression)

注意：selenium-webdriver通过findElement()\findElements()等find方法调用"By"对象来定位和查询元素。By类只是提供查询的方式进行分类。findElement返回一个元素对象否则抛出异常，findElements返回符合条件的元素List，如果不存在符合条件的就返回一个空的list。

使用className进行定位

当所定位的元素具有class属性的时候我们可以通过classname来定位该元素。

下面的例子定位了51.com首页上class为"username"的li。

```
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;

import org.openqa.selenium.By;
```

```
public class ByClassName {



    public static void main(String[] args) {
         WebDriver driver = new FirefoxDriver();
        driver.get("http://www.51.com");
         WebElement element = driver.findElement(By.className("username"));
         System.out.println(element.getTagName());

    }
}
```

输出结果：

```
li
```

使用id属性定位

51.com首页的帐号输入框的html代码如下：

```
<input id="passport_51_user" type="text" value="" tabindex="1" title="用户名/彩虹号/邮箱"
name="passport_51_user">
```

在下面的例子中我们用id定位这个输入框，并输出其title,借此也可以验证代码是否工作正常。

```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;

public class ByUserId {

        /**
```

```
         * @param args
         */
        public static void main(String[] args) {
                // TODO Auto-generated method stub
                WebDriver dr = new FirefoxDriver();
                dr.get("http://www.51.com");

                WebElement element = dr.findElement(By.id("passport_51_user"));
                System.out.println(element.getAttribute("title"));
        }

}
```

输出结果：

```
用户名/彩虹号/邮箱
```

使用name属性定位

51.com首页的帐号输入框的html代码如下：

```
<input id="passport_51_user" type="text" value="" tabindex="1" title="用户名/彩虹号/邮箱"
name="passport_51_user">
```

使用name定位

```
WebElement e = dr.findElement(By.name("passport_51_user"));
```

使用css属性定位

51.com首页的帐号输入框的html代码如下：

```
<input id="passport_51_user" type="text" value="" tabindex="1" title="用户名/彩虹号/邮箱"
name="passport_51_user">
```

使用css定位

```
WebElement e1 = dr.findElement(By.cssSelector("#passport_51_user"));
```

使用其他方式定位

在定位link元素的时候，可以使用link和link_text属性；

另外还可以使用tag_name属性定位任意元素；

定位多个元素

上面提到findElements()方法可以返回一个符合条件的元素List组。看下面例子。

```
import java.io.File;
import java.util.List;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxBinary;
import org.openqa.selenium.firefox.FirefoxDriver;

public class FindElementsStudy {

        /**
         * @author gongjf
         */
        public static void main(String[] args) {
                WebDriver  driver = new FirefoxDriver();
                driver.get("http://www.51.com");

                //定位到所有<input>标签的元素，然后输出他们的id
                List<WebElement> element = driver.findElements(By.tagName("input"));
```

```
                for (WebElement e : element){
                        System.out.println(e.getAttribute("id"));
                }

                driver.quit();
        }
}
```

输出结果：

```
passport_cookie_login
gourl
passport_login_from
passport_51_user
passport_51_password
passport_qq_login_2
btn_reg
passport_51_ishidden
passport_auto_login
```

上面的代码返回页面上所有input对象。很简单，没什么可说的。

层级定位

层级定位的思想是先定位父元素，然后再从父元素中精确定位出其我们需要选取的子元素。

层级定位一般的应用场景是无法直接定位到需要选取的元素，但是其父元素比较容易定位，通过定位父元素再遍历其子元素选择需要的目标元素，或者需要定位某个元素下所有的子元素。

下面的代码演示了如何使用层级定位class为"login"的div，然后再取得它下面的所有label，并打印出他们的文本

```
import java.io.File;
import java.util.List;
```

```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxBinary;
import org.openqa.selenium.firefox.FirefoxDriver;

public class LayerLocator {

        /**
         * @author gongjf
         */
        public static void main(String[] args) {

                WebDriver  driver = new FirefoxDriver();
                driver.get("http://www.51.com");

                //定位class为"login"的div，然后再取得它下面的所有label，并打印出他们的值
                WebElement element = driver.findElement(By.className("login"));
                 List<WebElement> el = element.findElements(By.tagName("label"));
                 for(WebElement e : el)
                System.out.println(e.getText());

        }

}
```

输出结果：

```
帐号：
密码：
隐身
下次自动登录
```

定位页面元素over了，下次写一下对frame的处理。

1.5 selenium webdriver学习（五）----------iframe的处理

发表时间: 2012-03-12 关键字: 如何定位frame中元素

有时候我们在定位一个页面元素的时候发现一直定位不了，反复检查自己写的定位器没有任何问题，代码也没有任何问题。这时你就要看一下这个页面元素是否在一个iframe中，这可能就是找不到的原因之一。如果你在一个default content中查找一个在iframe中的元素，那肯定是找不到的。反之你在一个iframe中查找另一个iframe元素或default content中的元素，那必然也定位不到。

selenium webdriver中提供了进入一个iframe的方法：

WebDriver org.openqa.selenium.WebDriver.TargetLocator.frame(String nameOrId)

也提供了一个返回default content的方法：

WebDriver org.openqa.selenium.WebDriver.TargetLocator.defaultContent()

这样使我们面对iframe时可以轻松应对。

以下面的html代码为例，我们看一下处现iframe。

```
main.html

<html>
    <head>
        <title>FrameTest</title>
    </head>
    <body>
        <div id = "id1">this is a div!</div>
        <iframe id = "frame"  frameborder="0" scrolling="no" style="left:0;position:absolute;"
    </body>
</html>



frame.html

<html>
```

```
    <head>
        <title>this is a frame!</title>
    </head>
    <body>
        <div id = "div1">this is a div , too!</div>
        <label>input:</label>
        <input id = "input1"></input>
    </body>
</html>
```

```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;

public class FameStudy {



        public static void main(String[] args) {
                WebDriver dr = new FirefoxDriver();
                String url = "\\Your\\Path\\to\\main.html";
                dr.get(url);

                //在default content定位id="id1"的div
                dr.findElement(By.id("id1"));

                //此时，没有进入到id="frame"的frame中时，以下两句会报错
                dr.findElement(By.id("div1"));//报错
                dr.findElement(By.id("input1"));//报错

                //进入id="frame"的frame中，定位id="div1"的div和id="input1"的输入框。
                dr.switchTo().frame("frame");
                dr.findElement(By.id("div1"));
                dr.findElement(By.id("input1"));

                //此时，没有跳出frame，如果定位default content中的元素也会报错。
```

```
                dr.findElement(By.id("id1"));//报错

                //跳出frame,进入default content;重新定位id="id1"的div
                dr.switchTo().defaultContent();
                dr.findElement(By.id("id1"));
        }

}
```

switch_to方法会new1个TargetLocator对象，使用该对象的frame方法可以将当前识别的”主体”移动到需要定位的frame上去。

1.6 selenium webdriver学习（六）----------如何得到弹出窗口

发表时间: 2012-03-12

在selenium 1.X里面得到弹出窗口是一件比较麻烦的事，特别是新开窗口没有id、name的时候。当时还整理了处理了几种方法，详见：http://seleniumcn.cn/read.php?tid=791 。在selenium webdriver中得到新开窗口相对简单的多，它无关新开窗口的id、name等属性。以下面的html为例：

test.html

```
<html>
    <head><title>Test Popup Window</title></head>
    <body>
        <a id = "51" href = "http://www.51.com/" target = "_blank">Let's go!</a>
    </body>
</html>
```

下面的代码演示了如何去得到弹出的新窗口

```
import java.util.Iterator;
import java.util.Set;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;

public class PopupWindowTest {


        /**
```

```
         * @author gongjf
         */
        public static void main(String[] args) {
                System.setProperty("webdriver.firefox.bin","D:\\Program Files\\Mozilla Firefox\
                WebDriver dr = new FirefoxDriver();
                String url ="\\Your\\Path\\to\\main.html";
                dr.get(url);
                dr.findElement(By.id("51")).click();
                //得到当前窗口的句柄
                String currentWindow = dr.getWindowHandle();
                //得到所有窗口的句柄
                Set<String> handles = dr.getWindowHandles();
                Iterator<String> it = handles.iterator();
                while(it.hasNext()){
                        if(currentWindow == it.next())  continue;
                        WebDriver       window = dr.switchTo().window(it.next());
                        System.out.println("title,url = "+window.getTitle()+","+window.getCurre
                }
        }

}
```

输出结果：

```
title,url = 51.com 真人配对玩游戏,http://www.51.com/
```

捕获或者说定位弹出窗口的关键在于获得弹出窗口的句柄。（句柄，我的理解是浏览器窗口的一个唯一标识，记得以前玩"按键精灵"也有这玩样。）

在上面的代码里，使用windowhandle方法来获取当前浏览器窗口的句柄，使用了windowhandles方法获取所有弹出的浏览器窗口的句柄，然后通过排除当前句柄的方法来得到新开窗口的句柄。

在获取新弹出窗口的句柄后，使用switchto.window(newwindow_handle)方法，将新窗口的句柄作为参数传入既可捕获到新窗口了。

如果想回到以前的窗口定位元素，那么再调用1次switch_to.window方法，传入之前窗口的句柄既可达到目的。

1.7 selenium webdriver学习（七）----------如何处理alert、confirm、prompt对话框

发表时间: 2012-03-12 关键字: alert, prompt, confirm, selenium, webdriver

alert、confirm、prompt这样的js对话框在selenium1.X时代也是难啃的骨头，常常要用autoit来帮助处理。

试用了一下selenium webdriver中处理这些对话框十分方便简洁。以下面html代码为例：

```
Dialogs.html
```

```
<html>

    <head>

        <title>Alert</title>

    </head>

    <body>

        <input id = "alert" value = "alert" type = "button" onclick = "alert('欢迎！请按确认继续
        <input id = "confirm" value = "confirm" type = "button" onclick = "confirm('确定吗？');'
        <input id = "prompt" value = "prompt" type = "button" onclick = "var name = prompt('请输

你的名字'); document.write(name) "/>


    </body>

</html>
```

以上html代码在页面上显示了三个按钮，点击他们分别弹出alert、confirm、prompt对话框。如果在prompt对话框中输入文字点击确定之后，将会刷新页面，显示出这些文字 。

selenium webdriver 处理这些弹层的代码如下：

```
import org.openqa.selenium.Alert;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;

public class DialogsStudy {

        /**
         * @author gongjf
         */
        public static void main(String[] args) {
                // TODO Auto-generated method stub
                System.setProperty("webdriver.firefox.bin","D:\\Program Files\\Mozilla Firefox\
                WebDriver dr = new FirefoxDriver();
                String url = "file:///C:/Documents and Settings/gongjf/桌面/selenium_test/Dialo
                dr.get(url);

                //点击第一个按钮，输出对话框上面的文字，然后叉掉
                dr.findElement(By.id("alert")).click();
                Alert alert = dr.switchTo().alert();
                String text = alert.getText();
                System.out.println(text);
                alert.dismiss();

                //点击第二个按钮，输出对话框上面的文字，然后点击确认
                dr.findElement(By.id("confirm")).click();
                Alert confirm = dr.switchTo().alert();
                String text1 = confirm.getText();
                System.out.println(text1);
                confirm.accept();

                //点击第三个按钮，输入你的名字，然后点击确认，最后
                dr.findElement(By.id("prompt")).click();
```

```
                Alert prompt = dr.switchTo().alert();
                String text2 = prompt.getText();
                System.out.println(text2);
                prompt.sendKeys("jarvi");
                prompt.accept();

        }

}
```

从以上代码可以看出dr.switchTo().alert();这句可以得到alert\confirm\prompt对话框的对象，然后运用其方法对它进行操作。对话框操作的主要方法有：

- getText()　得到它的文本值
- accept()　相当于点击它的"确认"
- dismiss()　相当于点击"取消"或者叉掉对话框
- sendKeys() 输入值，这个alert\confirm没有对话框就不能用了，不然会报错。

1.8 selenium webdriver学习（八）----------如何操作select下拉框

发表时间: 2012-03-12

下面我们来看一下selenium webdriver是如何来处理select下拉框的，以http://passport.51.com/reg2.5p这个页面为例。这个页面中有4个下拉框，下面演示4种选中下拉框选项的方法。select处理比较简单，直接看代码吧：）

```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.support.ui.Select;

public class SelectsStudy {

        /**
         * @author gongjf
         */
        public static void main(String[] args) {
                // TODO Auto-generated method stub
                System.setProperty("webdriver.firefox.bin","D:\\Program Files\\Mozilla Firefox\
                WebDriver dr = new FirefoxDriver();
                dr.get("http://passport.51.com/reg2.5p");

                //通过下拉列表中选项的索引选中第二项，即2011年
                Select selectAge = new Select(dr.findElement(By.id("User_Age")));
                selectAge.selectByIndex(2);

                //通过下拉列表中的选项的value属性选中"上海"这一项
                Select selectShen = new Select(dr.findElement(By.id("User_Shen")));
                selectShen.selectByValue("上海");

                //通过下拉列表中选项的可见文本选 中"浦东"这一项
                Select selectTown = new Select(dr.findElement(By.id("User_Town")));
                selectTown.selectByVisibleText("浦东");
```

```
                //这里只是想遍历一下下拉列表所有选项，用click进行选中选项
                Select selectCity = new Select(dr.findElement(By.id("User_City")));
                for(WebElement e : selectCity.getOptions())
                        e.click();
        }

}
```

从上面可以看出，对下拉框进行操作时首先要定位到这个下拉框，new 一个Selcet对象，然后对它进行操作。

1.9 selenium webdriver学习（九）----------如何操作cookies

发表时间: 2012-03-12

Web 测试中我们经常会接触到Cookies，一个Cookies主要属性有”所在域、name、value、有效日期和路径",下面来讲一下怎么操作Cookies。

```
import java.util.Set;

import org.openqa.selenium.Cookie;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;

public class CookiesStudy {

        /**
         * @author  gongjf
         */
        public static void main(String[] args) {
                // TODO Auto-generated method stub
                System.setProperty("webdriver.firefox.bin","D:\\Program Files\\Mozilla Firefox\
                WebDriver dr = new FirefoxDriver();
                dr.get("http://www.51.com");

                //增加一个name = "name",value="value"的cookie
                Cookie cookie = new Cookie("name", "value");
                dr.manage().addCookie(cookie);

                //得到当前页面下所有的cookies，并且输出它们的所在域、name、value、有效日期和路径
                Set<Cookie> cookies = dr.manage().getCookies();
                System.out.println(String.format("Domain -> name -> value -> expiry -> path"));
                for(Cookie c : cookies)
                        System.out.println(String.format("%s -> %s -> %s -> %s -> %s",
                                        c.getDomain(), c.getName(), c.getValue(),c.getExpiry(),


                //删除cookie有三种方法
```

```
            //第一种通过cookie的name
            dr.manage().deleteCookieNamed("CookieName");
            //第二种通过Cookie对象
            dr.manage().deleteCookie(cookie);
            //第三种全部删除
            dr.manage().deleteAllCookies();
        }
```

上面的代码首先在页面中增加了一个cookie,然后遍历页面的所有cookies，并输出他们的主要属性。最后就是三种删除cookie的方法。遍历cookies输出的结果：

```
Domain -> name -> value -> expiry -> path
.51.com -> FO_RFLP -> %7CaHR0cDovL3d3dy41MS5jb20v%7C%7C%7C -> null -> /
.51.com -> __utmz -> 67913429.1331544776.1.1.utmcsr=(direct)|utmccn=(direct)|utmcmd=(none) -> 1
www.51.com -> name -> value -> Tue Mar 12 17:33:00 CST 2030 -> /
www.51.com -> PHPSESSID -> 51d37fc72eb0ea66e4ef1971b688698b -> null -> /
.51.com -> __utma -> 67913429.453585250.1331544776.1331544776.1331544776.1 -> Wed Mar 12 17:32:
www.51.com -> www_cookie_adv -> 1 -> Mon Mar 12 18:32:55 CST 2012 -> /
.51.com -> __utmc -> 67913429 -> null -> /
www.51.com -> NSC_xxx -> 44595a553660 -> null -> /
.51.com -> __utmb -> 67913429.1.10.1331544776 -> Mon Mar 12 18:02:56 CST 2012 -> /
www.51.com -> www_jiaoyou_guide -> 0c83c0b5f569512d5a832bf0b4397a05 -> null -> /
```

1.10 selenium webdriver学习（十）----------如何把一个元素拖放到另一个元素里面

发表时间: 2012-03-13 关键字: 元素拖放, drag and drop

Q群里有时候会有人问，selenium webdriver怎么实现把一个元素拖放到另一个元素里面。这一节总一下元素的拖放。

下面这个页面是一个演示拖放元素的页面，你可以把左右页面中的条目拖放到右边的div框中。

http://koyoz.com/demo/html/drag-drop/drag-drop.html

现在来看看selenium webdriver是怎么实现drag and drop的吧。let‘s go！

```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.interactions.Actions;

public class DragAndDrop {

        /**
         * @author gongjf
         */
        public static void main(String[] args) {
                // TODO Auto-generated method stub
                System.setProperty("webdriver.firefox.bin","D:\\Program Files\\Mozilla Firefox\
                WebDriver dr = new FirefoxDriver();
                dr.get("http://koyoz.com/demo/html/drag-drop/drag-drop.html");

                //首先new出要拖入的页面元素对象和目标对象，然后进行拖入。
            WebElement  element = dr.findElement(By.id("item1"));
                WebElement  target = dr.findElement(By.id("drop"));
                (new Actions(dr)).dragAndDrop(element, target).perform();

                //利用循环把其它item也拖入
```

```
                String id="item" ;
                for(int i=2;i<=6;i++){
                        String item = id+i;
                        (new Actions(dr)).dragAndDrop(dr.findElement(By.id(item)), target).perf
                }
        }

}
```

代码很简单，需要注意的是(new Actions(dr)).dragAndDrop(element, target).perform();这句话中，dragAndDrop(element, target)这个方法是定义了“点击element元素对象，然后保持住，直到拖到目标元素对象里面才松开”这一系列动作的Actions,如果你不调用perform()方法，这个Actions是不会执行的。over！

1.11 selenium webdriver学习（十一）----------如何等待页面元素加载完成

发表时间: 2012-03-14 关键字: selenium webdriver, waitforcondition, 等待页面元素加载完成

web的自动化测试中，我们经常会遇到这样一种情况：当我们的程序执行时需要页面某个元素，而此时这个元素还未加载完成，这时我们的程序就会报错。怎么办？等待。等待元素出现后再进行对这个元素的操作。

在selenium-webdriver中我们用两种方式进行等待：明确的等待和隐性的等待。

明确的等待

明确的等待是指在代码进行下一步操作之前等待某一个条件的发生。最不好的情况是使用Thread.sleep()去设置一段确认的时间去等待。但为什么说最不好呢？因为一个元素的加载时间有长有短，你在设置sleep的时间之前要自己把握长短，太短容易超时，太长浪费时间。selenium webdriver提供了一些方法帮助我们等待正好需要等待的时间。利用WebDriverWait类和ExpectedCondition接口就能实现这一点。

下面的html代码实现了这样的一种效果：点击click按钮5秒钟后，页面上会出现一个红色的div块。我们需要写一段自动化脚本去捕获这个出现的div，然后高亮之。

```
Wait.html

<html>
    <head>
        <title>Set Timeout</title>
        <style>
            .red_box {background-color: red; width = 20%; height: 100px; border: none;}
        </style>
        <script>
            function show_div(){
```

```
                setTimeout("create_div()", 5000);
            }

            function create_div(){
                d = document.createElement('div');
                d.className = "red_box";
                document.body.appendChild(d);
            }
        </script>
    </head>
    <body>
        <button id = "b" onclick = "show_div()">click</button>
    </body>
</html>
```

下面的代码实现了高亮动态生成的div块的功能：

```
import org.openqa.selenium.By;
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.support.ui.ExpectedCondition;
import org.openqa.selenium.support.ui.WebDriverWait;


public class WaitForSomthing {

        /**
         * @author gongjf
         */
        public static void main(String[] args) {
                // TODO Auto-generated method stub
```

```
                System.setProperty("webdriver.firefox.bin","D:\\Program Files\\Mozilla Firefox\
                WebDriver dr = new FirefoxDriver();
                String url = "file:///C:/Documents and Settings/gongjf/桌面/selenium_test/Wait.
                dr.get(url);
                WebDriverWait wait = new WebDriverWait(dr,10);
                wait.until(new ExpectedCondition<WebElement>(){
                        @Override
                        public WebElement apply(WebDriver d) {
                                return d.findElement(By.id("b"));
                        }}).click();

                WebElement element = dr.findElement(By.cssSelector(".red_box"));
                ((JavascriptExecutor)dr).executeScript("arguments[0].style.border = \"5px solic

        }
}
```

上面的代码WebDriverWait类的构造方法接受了一个WebDriver对象和一个等待最长时间（10秒）。然后调用until方法，其中重写了ExpectedCondition接口中的apply方法，让其返回一个WebElement,即加载完成的元素，然后点击。默认情况下，WebDriverWait每500毫秒调用一次ExpectedCondition，直到有成功的返回，当然如果超过设定的值还没有成功的返回，将抛出异常。

隐性等待

隐性等待是指当要查找元素，而这个元素没有马上出现时，告诉WebDriver查询Dom一定时间。默认值是0,但是设置之后，这个时间将在WebDriver对象实例整个生命周期都起作用。上面的代码就变成了这样：

```
import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
```

```
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.support.ui.ExpectedCondition;
import org.openqa.selenium.support.ui.WebDriverWait;


public class WaitForSomthing {

        /**
         * @author gongjf
         */
        public static void main(String[] args) {
                // TODO Auto-generated method stub
                System.setProperty("webdriver.firefox.bin","D:\\Program Files\\Mozilla Firefox\
                WebDriver dr = new FirefoxDriver();

                //设置10秒
                dr.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

                String url = "file:///C:/Documents and Settings/gongjf/桌面/selenium_test/Wait.
                dr.get(url);
                 //注释掉原来的
                /*WebDriverWait wait = new WebDriverWait(dr,10);
                wait.until(new ExpectedCondition<WebElement>(){
                        @Override
                        public WebElement apply(WebDriver d) {
                                return d.findElement(By.id("b"));
                        }}).click();*/
                dr.findElement(By.id("b")).click();
                WebElement element = dr.findElement(By.cssSelector(".red_box"));
                ((JavascriptExecutor)dr).executeScript("arguments[0].style.border = \"5px solic

        }
}
```

两者选其一，第二种看起来一劳永逸呀。哈哈

1.12 selenium webdriver学习（十二）----------如何利用selenium-webdriver截图

发表时间: 2012-03-26

在自动化测试中常常会用到截图功能。最近用了一下selenium-webdriver的截图功能还算不错，可以截取页面全图，不管页面有多长。

下面的代码演示了如何使用webdriver进行截图：

```
import java.io.File;
import java.io.IOException;

import org.apache.commons.io.FileUtils;
import org.openqa.selenium.OutputType;
import org.openqa.selenium.TakesScreenshot;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
public class ShotScreen {

        /**
         * @author gongjf
         * @throws IOException
         * @throws InterruptedException
         */
        public static void main(String[] args) throws IOException, InterruptedException {

                System.setProperty("webdriver.firefox.bin","D:\\Program Files\\Mozilla Firefox\
                WebDriver dr = new FirefoxDriver();
                dr.get("http://www.51.com");

                //这里等待页面加载完成
                Thread.sleep(5000);
                //下面代码是得到截图并保存在D盘下
                File screenShotFile = ((TakesScreenshot)dr).getScreenshotAs(OutputType.FILE);
                FileUtils.copyFile(screenShotFile, new File("D:/test.png"));}}
```

看了一下OutputType接口和TakesScreenshot接口，吐槽一下，貌似这两个接口不是同一个开发写的或者注释没有更新怎么的。在OutputType里面的注释说：

```
/**
 * Defines the output type for a screenshot. See org.openqa.selenium.Screenshot for usage and
 * examples.
...
```

然后在那找了半天的org.openqa.selenium.Screenshot 接口，晕，后来想应该是org.openqa.selenium.TakesScreenshot。

在TakesScreenshot里有如下注释：

```
 /**
  * Capture the screenshot and store it in the specified location.
  *
  * <p>For WebDriver extending TakesScreenshot, this makes a best effort
  * depending on the browser to return the following in order of preference:
  * <ul>
  *   <li>Entire page</li>
  *   <li>Current window</li>
  *   <li>Visible portion of the current frame</li>
  *   <li>The screenshot of the entire display containing the browser</li>
  * </ul>
  *
  * <p>For WebElement extending TakesScreenshot, this makes a best effort
  * depending on the browser to return the following in order of preference:
  *   - The entire content of the HTML element
  *   - The visisble portion of the HTML element
  *
  * @param <X> Return type for getScreenshotAs.
  * @param target target type, @see OutputType
  * @return Object in which is stored information about the screenshot.
```

```
    * @throws WebDriverException on failure.
    */
```

试了一下截取Webelement最终发现WebElement接口没有实现这个类。搞了半天也只是会了截取页面的全图。截取当前的frame也截取的页面全图。难道这个功能没有完善，好吧，这样说自我安慰一下。

selenium-webdriver 面向接口编程，找一个需要的功能还真是挺难的。

1.13 selenium webdriver学习（十三）----------如何利用Actions类模拟鼠标和键盘的操作

发表时间: 2012-03-29

在selenium webdriver学习（十）----------如何把一个元素拖放到另一个元素里面 的时候，用到了一个Actions类。这一节主要分析一下这个Actions类。

这个actions类，主要定义了一些模拟用户的鼠标mouse，键盘keyboard操作。对于这些操作，使用perform()方法进行执行。

actions类可以完成单一的操作，也可以完成几个操作的组合。

单一的操作

单一的操作是指鼠标和键盘的一个操作。如鼠标左键按下、弹起或输入一个字符串等。

前面涉及到鼠标键盘操作的一些方法，都可以使用actions类中的方法实现，比如：click，sendkeys。

```
WebElement  element = dr.findElement(By.id("test"));
 WebElement  element1 = dr.findElement(By.id("test1"));
element.sendKeys("test");
element1.click;
```

用Actions类就可以这样实现：

```
//新建一个action
Actions action=new Actions(driver);
//操作
WebElement element=dr.findElement(By.id("test"));
WebElement element1=dr.findElement(By.id("su"));
action.sendKeys(element,"test").perform();
action.moveToElement(element1);
action.click().perform();
```

看起来用Actions类实现click和sendKeys有点烦索

组合操作

组合操作就是几个动作连在一起进行操作。如对一个元素的拖放。

```
(new Actions(dr)).dragAndDrop(dr.findElement(By.id(item)), target).perform();
```

可以直接调用dragAndDrip()方法，也可以像下面濱示的一样把几个操作放一起实现

```
Action dragAndDrop = builder.clickAndHold(someElement)
       .moveToElement(otherElement)
       .release(otherElement)
       .build().perform();
```

其他鼠标或键盘操作方法可以具体看一下API里面的org.openqa.selenium.interactions.Actions类

1.14 selenium webdriver学习（十四）----------如何处理table

发表时间: 2012-04-07 关键字: selenium, webdriver, table, table 操作

以前在selenium RC 里面有一个getTable方法，是得到一个单元格中的文本。其详细描述如下：

```
/** Gets the text from a cell of a table. The cellAddress syntax tableLocator.row.column
, where row and column start at 0.
@param tableCellAddress a cell address, e.g. "foo.1.4"
@return the text from the specified cell
*/
String getTable(String tableCellAddress);
```

就是传入一个参数，这个参数的格式必须是tableLocator.row.column，如"foo.1.4"，foo用于得到table对象，1.4代表在table里第1行第4列。行、列从0开始。

在selenium webdriver里，没有这样的方法，也就是说没有专门操作table的类。但我们可以自己封闭一个，这并不难。以上面的getTable方法为例，我们自己也可以创建这样功能的一个方法。

```
public String getCellText(By by,String tableCellAddress)
```

我叫它getCellText,它有两个参数，第一个是By对象用于得到table对象， tableCellAddress 如"1.4",代表在table里第1行第4列。行、列从0开始。

以下面html代码为例：

```
<html>
    <head>
        <title>Table</title>
```

```
    </head>
    <body>
        <table border="1" id="myTable">
                        <tr>
                                <th>Heading(row 0 ,cell 0)</th>
                                <th>Another Heading(row 0 ,cell 1)</th>
                                <th>Another Heading(row 0 ,cell 2)</th>
                        </tr>
                        <tr>
                                <td>row 1, cell 0</td>
                                <td>row 1, cell 1</td>
                                <td>row 1, cell 2</td>
                        </tr>
                        <tr>
                                <td>row 2, cell 0</td>
                                <td>row 2, cell 1</td>
                                <td>row 2, cell 2</td>
                        </tr>
                </table>
    </body>
</html>
```

示例代码如下：

```
import java.util.List;
import org.openqa.selenium.By;
import org.openqa.selenium.NoSuchElementException;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
```

```
public class Table {

	/**
	 * @author gongjf
	 */
	private WebDriver driver;
	Table(WebDriver driver){
		this.driver = driver;
	}

	/** 从一个table的单元格中得到文本值. 参数tableCellAddress的格式为
	row.column, 行列从0开始.
	@param by  用于得到table对象
	@param tableCellAddress 一个单元格地址, 如. "1.4"
	@return 从一个table的单元格中得到文本值
	*/
	public String getCellText(By by,String tableCellAddress) {
		//得到table元素对象
		WebElement table = driver.findElement(by);
		//对所要查找的单元格位置字符串进行分解，得到其对应行、列。
		int index = tableCellAddress.trim().indexOf('.');
		int row =  Integer.parseInt(tableCellAddress.substring(0, index));
		int cell = Integer.parseInt(tableCellAddress.substring(index+1));
		//得到table表中所有行对象，并得到所要查询的行对象。
		 List<WebElement> rows = table.findElements(By.tagName("tr"));
		 WebElement theRow = rows.get(row);
		 //调用getCell方法得到对应的列对象，然后得到要查询的文本。
		 String text = getCell(theRow, cell).getText();
		 return text;
	}
	private WebElement getCell(WebElement Row,int cell){
		 List<WebElement> cells;
		 WebElement target = null;
		 //列里面有"<th>"、"<td>"两种标签，所以分开处理。
		 if(Row.findElements(By.tagName("th")).size()>0){
			cells = Row.findElements(By.tagName("th"));
```

```
                        target = cells.get(cell);
                }
                if(Row.findElements(By.tagName("td")).size()>0){
                        cells = Row.findElements(By.tagName("td"));
                        target = cells.get(cell);
                }
               return target;

        }


        public static void main(String[] args) {
                 WebDriver driver;
                 System.setProperty("webdriver.firefox.bin","D:\\Program Files\\Mozilla Firefox
                 driver = new FirefoxDriver();
                 driver.get("file:///C:/Documents and Settings/Gongjf/桌面/selenium_test/table.

                 Table table = new Table(driver);
                 By by = By.id("myTable");
                 String address = "0.2";

                 System.out.println(table.getCellText(by, address));


        }

}
```

运行代码将输出

```
Another Heading(row 0 ,cell 2)
```

ps: 这里我只是以得到一个table中单元格的文本为例，但是从代码可以看出，对table的基本操作都有涉及到。有用到的同学可以自己包装一个完整的table类。

1.15 selenium webdriver学习（十五）----------如何处理FirefoxProfile

发表时间: 2012-04-10 关键字: selenium 2, selenium webdriver, firefox profile

这一节主要涉及 selenium webdriver处理Firefox profile的一些知识。

什么是Firefox profile

要了解Firefox profile请访问这里，它详细解绍了Firefox proflie。在Firefox里，如何管理Firefox profile 请访问这里。看完它们，相信你对Firefox profile会有所了解。好了，必备的知识准备完了，让我们来看看selenium webdriver 是怎么操作Firefox profile的吧。

设置profile中的一个preference

```
FirefoxProfile profile = new FirefoxProfile();
profile.setPreference("aaa", "bbbb");
WebDriver driver = new FirefoxDriver(profile);
```

以上代码在Firefox Profile文件中设置一个名aaa，值为bbb的preference.(ps:这个preference只是一个举例，没有任何意义。要看firefox profile有哪些preference,可以在firefox浏览器地址栏中输入:about:config). 代码运行后，在firefox浏览器地址栏中输入:about:config，可以看到它。

启用已经存在的profile

首先来了解一下为什么要已经存在的profile，其中一个原因是已经存在的profile里面保存有cookie等信息，可以保持用户的登录状态。

启动已经存在的profile，因profile不同而有两种方法。一种是如果这个profile使用firefox配置管理器（Firefox's profile manager）而已经存在了。我们用下面的方法：

```
ProfilesIni allProfiles = new ProfilesIni();
FirefoxProfile profile = allProfiles.getProfile("WebDriver");
WebDriver driver = new FirefoxDriver(profile);
```

另一种是没有在自己的firefox里面注册过的，比如从另一台机子中的firefox得到的，我们可以用下面的代码：

```
File profileDir = new File("path/to/your/profile");
FirefoxProfile profile = new FirefoxProfile(profileDir);
WebDriver driver = new FirefoxDriver(profile);
```

临时指定插件

有时我们需要临时让启动的firefox带一个插件，如firebug,来定位问题等。首先我们要下载这个插件的xpi安装包。剩下的就让selenium webdriver 来完成，如下：

```
File file = new File("path/to/your/firebug-1.8.1.xpi");
```

```
   FirefoxProfile firefoxProfile = new FirefoxProfile();
   firefoxProfile.addExtension(file);
   firefoxProfile.setPreference("extensions.firebug.currentVersion", "1.8.1"); //避免启动画面
   WebDriver driver = new FirefoxDriver(firefoxProfile);
```

这样启动的firefox中就安装了插件firebug.

启用默认情况下被firefox禁用的功能

以本地事件例，很简单直接设置为true就可以了。

```
FirefoxProfile profile = new FirefoxProfile();
profile.setEnableNativeEvents(true);
WebDriver driver = new FirefoxDriver(profile);
```

其它设置见selenium webdriver API中的org.openqa.selenium.firefox.FirefoxProfile.

启用firefox代理

这个更简单，直接上代码了。

```
String PROXY = "localhost:8080";//如果不是本机，localhost替换成IP地址

org.openqa.selenium.Proxy proxy = new org.openqa.selenium.Proxy();
proxy.setHttpProxy(PROXY)
     .setFtpProxy(PROXY)
     .setSslProxy(PROXY);
DesiredCapabilities cap = new DesiredCapabailities();
cap.setPreference(CapabilityType.PROXY, proxy);
WebDriver driver = new FirefoxDriver(cap);
```

over ！

1.16 selenium webdriver学习（十六）----------用selenium webdriver实现selenium RC中的类似的方法

发表时间: 2012-05-11

最近想总结一下学习selenium webdriver的情况，于是就想用selenium webdriver里面的方法来实现selenium RC中操作的一些方法。目前封装了一个ActionDriverHelper类，来实现RC中Selenium.java和DefaultSelenium.java中的方法。有一些方法还没有实现，写的方法大多没有经过测试，仅供参考。代码如下：

```
package core;

import java.io.File;
import java.io.IOException;
import java.util.HashSet;
import java.util.List;
import java.util.Set;
import java.util.concurrent.TimeUnit;

import org.apache.commons.io.FileUtils;
import org.openqa.selenium.By;
import org.openqa.selenium.Cookie;
import org.openqa.selenium.Dimension;
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.Keys;
import org.openqa.selenium.NoSuchElementException;
import org.openqa.selenium.OutputType;
import org.openqa.selenium.Point;
import org.openqa.selenium.TakesScreenshot;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.WebDriver.Timeouts;
import org.openqa.selenium.interactions.Actions;
import org.openqa.selenium.support.ui.Select;

public class ActionDriverHelper {
                protected WebDriver driver;
```

```
public ActionDriverHelper(WebDriver driver){
        this.driver = driver ;
}



public void click(By by) {
        driver.findElement(by).click();
}

public void doubleClick(By by){
        new Actions(driver).doubleClick(driver.findElement(by)).perform();
}

public void contextMenu(By by) {
        new Actions(driver).contextClick(driver.findElement(by)).perform();
}

public void clickAt(By by,String coordString) {
        int index = coordString.trim().indexOf(',');
        int xOffset = Integer.parseInt(coordString.trim().substring(0, index));
        int yOffset = Integer.parseInt(coordString.trim().substring(index+1));
        new Actions(driver).moveToElement(driver.findElement(by), xOffset, yOff
}

public void doubleClickAt(By by,String coordString){
        int index = coordString.trim().indexOf(',');
        int xOffset = Integer.parseInt(coordString.trim().substring(0, index));
        int yOffset = Integer.parseInt(coordString.trim().substring(index+1));
        new Actions(driver).moveToElement(driver.findElement(by), xOffset, yOff
                                                .doubleClick(driver.findElement
                                                .perform();
}

public void contextMenuAt(By by,String coordString) {
        int index = coordString.trim().indexOf(',');
        int xOffset = Integer.parseInt(coordString.trim().substring(0, index));
        int yOffset = Integer.parseInt(coordString.trim().substring(index+1));
```

```
        new Actions(driver).moveToElement(driver.findElement(by), xOffset, yOff
                                                .contextClick(driver.findElemer
                                                .perform();
    }

    public void fireEvent(By by,String eventName) {
        System.out.println("webdriver 不建议使用这样的方法，所以没有实现。");
    }

    public void focus(By by) {
        System.out.println("webdriver 不建议使用这样的方法，所以没有实现。");
    }

    public void keyPress(By by,Keys theKey) {
        new Actions(driver).keyDown(driver.findElement(by), theKey).release().p
    }

    public void shiftKeyDown() {
        new Actions(driver).keyDown(Keys.SHIFT).perform();
    }

    public void shiftKeyUp() {
        new Actions(driver).keyUp(Keys.SHIFT).perform();
    }

    public void metaKeyDown() {
        new Actions(driver).keyDown(Keys.META).perform();
    }

    public void metaKeyUp() {
        new Actions(driver).keyUp(Keys.META).perform();
    }

    public void altKeyDown() {
        new Actions(driver).keyDown(Keys.ALT).perform();
    }
```

```
public void altKeyUp() {
        new Actions(driver).keyUp(Keys.ALT).perform();
}

public void controlKeyDown() {
        new Actions(driver).keyDown(Keys.CONTROL).perform();
}

public void controlKeyUp() {
        new Actions(driver).keyUp(Keys.CONTROL).perform();
}

public void KeyDown(Keys theKey) {
        new Actions(driver).keyDown(theKey).perform();
}
public void KeyDown(By by,Keys theKey){
        new Actions(driver).keyDown(driver.findElement(by), theKey).perform();
}

public void KeyUp(Keys theKey){
        new Actions(driver).keyUp(theKey).perform();
}

public void KeyUp(By by,Keys theKey){
        new Actions(driver).keyUp(driver.findElement(by), theKey).perform();
}

public void mouseOver(By by) {
        new Actions(driver).moveToElement(driver.findElement(by)).perform();
}

public void mouseOut(By by) {
        System.out.println("没有实现！");
        //new Actions(driver).moveToElement((driver.findElement(by)), -10, -10)
}

public void mouseDown(By by) {
```

```
        new Actions(driver).clickAndHold(driver.findElement(by)).perform();
    }

    public void mouseDownRight(By by) {
        System.out.println("没有实现！");
    }

    public void mouseDownAt(By by,String coordString) {
        System.out.println("没有实现！");
    }

    public void mouseDownRightAt(By by,String coordString) {
        System.out.println("没有实现！");
    }

    public void mouseUp(By by) {
        System.out.println("没有实现！");
    }

    public void mouseUpRight(By by) {
        System.out.println("没有实现！");
    }

    public void mouseUpAt(By by,String coordString) {
        System.out.println("没有实现！");
    }

    public void mouseUpRightAt(By by,String coordString) {
        System.out.println("没有实现！");
    }

    public void mouseMove(By by) {
        new Actions(driver).moveToElement(driver.findElement(by)).perform();
    }

    public void mouseMoveAt(By by,String coordString) {
        int index = coordString.trim().indexOf(',');
```

```
            int xOffset = Integer.parseInt(coordString.trim().substring(0, index));
            int yOffset = Integer.parseInt(coordString.trim().substring(index+1));
            new Actions(driver).moveToElement(driver.findElement(by),xOffset,yOffse
    }
    public void type(By by, String testdata) {
            driver.findElement(by).clear();
            driver.findElement(by).sendKeys(testdata);
    }

    public void typeKeys(By by, Keys key) {
            driver.findElement(by).sendKeys(key);
    }
    public void setSpeed(String value) {
         System.out.println("The methods to set the execution speed in WebDriver we
    }

    public String getSpeed() {
            System.out.println("The methods to set the execution speed in WebDriver
            return null;

    }
    public void check(By by) {
            if(!isChecked(by))
                    click(by);
    }

    public void uncheck(By by) {
            if(isChecked(by))
                    click(by);
    }

    public void select(By by,String optionValue) {
            new Select(driver.findElement(by)).selectByValue(optionValue);
    }

    public void select(By by,int index) {
            new Select(driver.findElement(by)).selectByIndex(index);
```

```
    }

    public void addSelection(By by,String optionValue) {
        select(by,optionValue);
    }
    public void addSelection(By by,int index) {
        select(by,index);
    }

    public void removeSelection(By by,String value) {
        new Select(driver.findElement(by)).deselectByValue(value);
    }

    public void removeSelection(By by,int index) {
        new Select(driver.findElement(by)).deselectByIndex(index);
    }

    public void removeAllSelections(By by) {
        new Select(driver.findElement(by)).deselectAll();
    }

    public void submit(By by) {
        driver.findElement(by).submit();
    }

    public void open(String url) {
        driver.get(url);
    }

    public void openWindow(String url,String handler) {
        System.out.println("方法没有实现！");
    }

    public void selectWindow(String handler) {
        driver.switchTo().window(handler);
    }
```

```
public String getCurrentHandler(){
        String currentHandler = driver.getWindowHandle();
        return currentHandler;
}

public String getSecondWindowHandler(){
        Set<String> handlers = driver.getWindowHandles();
        String reHandler = getCurrentHandler();
        for(String handler : handlers){
                if(reHandler.equals(handler))  continue;
                reHandler = handler;
        }
        return reHandler;
}

public void selectPopUp(String handler) {
        driver.switchTo().window(handler);
}

public void selectPopUp() {
        driver.switchTo().window(getSecondWindowHandler());
}

public void deselectPopUp() {
        driver.switchTo().window(getCurrentHandler());
}

public void selectFrame(int index) {
        driver.switchTo().frame(index);
}

public void selectFrame(String str) {
        driver.switchTo().frame(str);
}

public void selectFrame(By by) {
        driver.switchTo().frame(driver.findElement(by));
```

```
}
public void waitForPopUp(String windowID,String timeout) {
        System.out.println("没有实现");
}
public void accept(){
        driver.switchTo().alert().accept();
}
public void dismiss(){
        driver.switchTo().alert().dismiss();
}
public void chooseCancelOnNextConfirmation() {
        driver.switchTo().alert().dismiss();
}

public void chooseOkOnNextConfirmation() {
        driver.switchTo().alert().accept();
}

public void answerOnNextPrompt(String answer) {
        driver.switchTo().alert().sendKeys(answer);
}

public void goBack() {
        driver.navigate().back();
}

public void refresh() {
        driver.navigate().refresh();
}

public void forward() {
        driver.navigate().forward();
}

public void to(String urlStr){
        driver.navigate().to(urlStr);
}
```

```
public void close() {
        driver.close();
}

public boolean isAlertPresent() {
        Boolean b = true;
        try{
                driver.switchTo().alert();
        }catch(Exception e){
                b = false;
        }
        return b;
}

public boolean isPromptPresent() {
        return isAlertPresent();
}

public boolean isConfirmationPresent() {
        return isAlertPresent();
}

public String getAlert() {
        return driver.switchTo().alert().getText();
}

public String getConfirmation() {
        return getAlert();
}

public String getPrompt() {
        return getAlert();
}

public String getLocation() {
        return driver.getCurrentUrl();
```

```
}

public String getTitle(){
        return driver.getTitle();
}

public String getBodyText() {
        String str = "";
        List<WebElement> elements = driver.findElements(By.xpath("//body//*[cor
        for(WebElement e : elements){
                str += e.getText()+" ";
        }
         return str;
}

public String getValue(By by) {
        return driver.findElement(by).getAttribute("value");
}

public String getText(By by) {
        return driver.findElement(by).getText();
}

public void highlight(By by) {
        WebElement element = driver.findElement(by);
        ((JavascriptExecutor)driver).executeScript("arguments[0].style.border =
}

public Object getEval(String script,Object... args) {
        return ((JavascriptExecutor)driver).executeScript(script,args);
}
public Object getAsyncEval(String script,Object... args){
        return  ((JavascriptExecutor)driver).executeAsyncScript(script, args);
}
public boolean isChecked(By by) {
        return driver.findElement(by).isSelected();
}
```

```
public String getTable(By by,String tableCellAddress) {
        WebElement table = driver.findElement(by);
        int index = tableCellAddress.trim().indexOf('.');
        int row =  Integer.parseInt(tableCellAddress.substring(0, index));
        int cell = Integer.parseInt(tableCellAddress.substring(index+1));
         List<WebElement> rows = table.findElements(By.tagName("tr"));
         WebElement theRow = rows.get(row);
         String text = getCell(theRow, cell);
         return text;
}
private String getCell(WebElement Row,int cell){
         List<WebElement> cells;
         String text = null;
         if(Row.findElements(By.tagName("th")).size()>0){
                cells = Row.findElements(By.tagName("th"));
                text = cells.get(cell).getText();
         }
         if(Row.findElements(By.tagName("td")).size()>0){
                cells = Row.findElements(By.tagName("td"));
                text = cells.get(cell).getText();
         }
        return text;

}

public String[] getSelectedLabels(By by) {
                Set<String> set = new HashSet<String>();
            List<WebElement> selectedOptions = new Select(driver.findElement(by

            for(WebElement e : selectedOptions){
                set.add(e.getText());
            }
        return set.toArray(new String[set.size()]);
}

public String getSelectedLabel(By by) {
        return getSelectedOption(by).getText();
```

```
}

public String[] getSelectedValues(By by) {
        Set<String> set = new HashSet<String>();
    List<WebElement> selectedOptions = new Select(driver.findElement(by))

    for(WebElement e : selectedOptions){
        set.add(e.getAttribute("value"));
    }
return set.toArray(new String[set.size()]);
}

public String getSelectedValue(By by) {
        return getSelectedOption(by).getAttribute("value");
}

public String[] getSelectedIndexes(By by) {
        Set<String> set = new HashSet<String>();
    List<WebElement> selectedOptions = new Select(driver.findElement(by))

   List<WebElement> options = new Select(driver.findElement(by)).getOptions();
    for(WebElement e : selectedOptions){
        set.add(String.valueOf(options.indexOf(e)));
    }
    return set.toArray(new String[set.size()]);
}

public String getSelectedIndex(By by) {
        List<WebElement> options = new Select(driver.findElement(by)).getOptior
        return String.valueOf(options.indexOf(getSelectedOption(by)));
}

public String[] getSelectedIds(By by) {
        Set<String> ids = new HashSet<String>();
        List<WebElement> options = new Select(driver.findElement(by)).getOptior
        for(WebElement option : options){
                if(option.isSelected()) {
```

```
                        ids.add(option.getAttribute("id")) ;
                }
        }
        return ids.toArray(new String[ids.size()]);
}

public String getSelectedId(By by) {
        return getSelectedOption(by).getAttribute("id");
}
private WebElement getSelectedOption(By by){
        WebElement selectedOption = null;
        List<WebElement> options = new Select(driver.findElement(by)).getOptior
        for(WebElement option : options){
                if(option.isSelected()) {
                        selectedOption = option;
                }
        }
        return selectedOption;
}

public boolean isSomethingSelected(By by) {
        boolean b = false;
        List<WebElement> options = new Select(driver.findElement(by)).getOptior
        for(WebElement option : options){
                if(option.isSelected()) {
                        b = true ;
                        break;
                }
        }
        return b;
}

public String[] getSelectOptions(By by) {
        Set<String> set = new HashSet<String>();
    List<WebElement> options = new Select(driver.findElement(by)).getOptions();
    for(WebElement e : options){
        set.add(e.getText());
```

```
    }
return set.toArray(new String[set.size()]);
}
public String getAttribute(By by,String attributeLocator) {
        return driver.findElement(by).getAttribute(attributeLocator);
}

public boolean isTextPresent(String pattern) {
        String Xpath= "//*[contains(text(),\'"+pattern+"\')]" ;
        try {
                driver.findElement(By.xpath(Xpath));
return true;
        } catch (NoSuchElementException e) {
return false;
        }
}

public boolean isElementPresent(By by) {
        return driver.findElements(by).size() > 0;
}

public boolean isVisible(By by) {
        return driver.findElement(by).isDisplayed();
}

public boolean isEditable(By by) {
        return driver.findElement(by).isEnabled();
}
public List<WebElement> getAllButtons() {
        return driver.findElements(By.xpath("//input[@type='button']"));
}

public List<WebElement> getAllLinks() {
        return driver.findElements(By.tagName("a"));
}

public List<WebElement> getAllFields() {
```

```
        return driver.findElements(By.xpath("//input[@type='text']"));
    }

    public String[] getAttributeFromAllWindows(String attributeName) {
        System.out.println("不知道怎么实现");
        return null;
    }
    public void dragdrop(By by,String movementsString) {
        dragAndDrop(by, movementsString);
    }
    public void dragAndDrop(By by,String movementsString) {
        int index = movementsString.trim().indexOf('.');
        int xOffset = Integer.parseInt(movementsString.substring(0, index));
        int yOffset = Integer.parseInt(movementsString.substring(index+1));
        new Actions(driver).clickAndHold(driver.findElement(by)).moveByOffset(x
    }
    public void setMouseSpeed(String pixels) {
        System.out.println("不支持");
    }

    public Number getMouseSpeed() {
        System.out.println("不支持");
        return null;
    }
    public void dragAndDropToObject(By source,By target) {
        new Actions(driver).dragAndDrop(driver.findElement(source), driver.finc
    }

    public void windowFocus() {
        driver.switchTo().defaultContent();
    }

    public void windowMaximize() {
        driver.manage().window().setPosition(new Point(0,0));
        java.awt.Dimension screenSize = java.awt.Toolkit.getDefaultToolkit().ge
        Dimension dim = new Dimension((int) screenSize.getWidth(), (int) screer
        driver.manage().window().setSize(dim);
```

```
        }

        public String[] getAllWindowIds() {
                System.out.println("不能实现！");
                return null;
        }

        public String[] getAllWindowNames() {
                System.out.println("不能实现！");
                return null;
        }

        public String[] getAllWindowTitles() {
                Set<String> handles = driver.getWindowHandles();
                Set<String> titles = new HashSet<String>();
                for(String handle : handles){
                        titles.add(driver.switchTo().window(handle).getTitle());
                }
                return titles.toArray(new String[titles.size()]);
        }
        public String getHtmlSource() {
                return driver.getPageSource();
        }

        public void setCursorPosition(String locator,String position) {
                System.out.println("没能实现！");
        }

        public Number getElementIndex(String locator) {
                System.out.println("没能实现！");
                return null;
        }

        public Object isOrdered(By by1,By by2) {
                System.out.println("没能实现！");
                return null;
        }
```

```
public Number getElementPositionLeft(By by) {
        return driver.findElement(by).getLocation().getX();
}

public Number getElementPositionTop(By by) {
        return driver.findElement(by).getLocation().getY();
}

public Number getElementWidth(By by) {
        return driver.findElement(by).getSize().getWidth();
}

public Number getElementHeight(By by) {
        return driver.findElement(by).getSize().getHeight();
}

public Number getCursorPosition(String locator) {
        System.out.println("没能实现！");
        return null;
}

public String getExpression(String expression) {
        System.out.println("没能实现！");
        return null;
}

public Number getXpathCount(By xpath) {
        return driver.findElements(xpath).size();
}

public void assignId(By by,String identifier) {
        System.out.println("不想实现！");
}

/*public void allowNativeXpath(String allow) {
        commandProcessor.doCommand("allowNativeXpath", new String[] {allow,});
```

```
	}*/

	/*public void ignoreAttributesWithoutValue(String ignore) {
		commandProcessor.doCommand("ignoreAttributesWithoutValue", new String[]

	}*/

	public void waitForCondition(String script,String timeout,Object... args) {
		Boolean b = false;
		int time = 0;
		while(time <= Integer.parseInt(timeout)){
			b = (Boolean) ((JavascriptExecutor)driver).executeScript(script
			if(b==true) break;
			try {
				Thread.sleep(1000);
			} catch (InterruptedException e) {
				// TODO Auto-generated catch block
				e.printStackTrace();
			}
			time += 1000;
		}
	}

	public void setTimeout(String timeout) {
		driver.manage().timeouts().implicitlyWait(Integer.parseInt(timeout), Ti
	}

	public void waitForPageToLoad(String timeout) {
		driver.manage().timeouts().pageLoadTimeout(Integer.parseInt(timeout), T
	}

	public void waitForFrameToLoad(String frameAddress,String timeout) {
		/*driver.switchTo().frame(frameAddress)
							.manage()
							.timeouts()
							.pageLoadTimeout(Integer.parseI
	}
```

```
public String getCookie() {
        String cookies = "";
        Set<Cookie> cookiesSet = driver.manage().getCookies();
        for(Cookie c : cookiesSet){
                cookies += c.getName()+"="+c.getValue()+";";
                }
        return cookies;
}

public String getCookieByName(String name) {
        return driver.manage().getCookieNamed(name).getValue();
}

public boolean isCookiePresent(String name) {
        boolean b = false ;
        if(driver.manage().getCookieNamed(name) != null || driver.manage().getC
                b = true;
        return b;
}

public void createCookie(Cookie c) {

        driver.manage().addCookie(c);
}

public void deleteCookie(Cookie c) {
        driver.manage().deleteCookie(c);
}

public void deleteAllVisibleCookies() {
        driver.manage().getCookieNamed("fs").isSecure();
}

/*public void setBrowserLogLevel(String logLevel) {

}*/
```

```
/*public void runScript(String script) {
        commandProcessor.doCommand("runScript", new String[] {script,});
}*/

/*public void addLocationStrategy(String strategyName,String functionDefinitior
        commandProcessor.doCommand("addLocationStrategy", new String[] {strateg
}*/

public void captureEntirePageScreenshot(String filename) {
        File screenShotFile = ((TakesScreenshot)driver).getScreenshotAs(OutputT
        try {
                FileUtils.copyFile(screenShotFile, new File(filename));
        } catch (IOException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
        }
}

/*public void rollup(String rollupName,String kwargs) {
        commandProcessor.doCommand("rollup", new String[] {rollupName,kwargs,})
}

public void addScript(String scriptContent,String scriptTagId) {
        commandProcessor.doCommand("addScript", new String[] {scriptContent,scr
}

public void removeScript(String scriptTagId) {
        commandProcessor.doCommand("removeScript", new String[] {scriptTagId,})
}

public void useXpathLibrary(String libraryName) {
        commandProcessor.doCommand("useXpathLibrary", new String[] {libraryName
}

public void setContext(String context) {
        commandProcessor.doCommand("setContext", new String[] {context,});
```

```
            }*/

            /*public void attachFile(String fieldLocator,String fileLocator) {
                    commandProcessor.doCommand("attachFile", new String[] {fieldLocator,fil
            }*/

            /*public void captureScreenshot(String filename) {
                    commandProcessor.doCommand("captureScreenshot", new String[] {filename,
            }*/

            public String captureScreenshotToString() {
                     String screen = ((TakesScreenshot)driver).getScreenshotAs(OutputType.B
                     return screen;
            }

        /* public String captureNetworkTraffic(String type) {
            return commandProcessor.getString("captureNetworkTraffic", new String[] {type})
         }
*/
         /*public void addCustomRequestHeader(String key, String value) {
            commandProcessor.getString("addCustomRequestHeader", new String[] {key, value})
         }*/

         /*public String captureEntirePageScreenshotToString(String kwargs) {
                    return commandProcessor.getString("captureEntirePageScreenshotToString"
            }*/

            public void shutDown() {
                    driver.quit();
            }

            /*public String retrieveLastRemoteControlLogs() {
                    return commandProcessor.getString("retrieveLastRemoteControlLogs", new
            }*/

            public void keyDownNative(Keys keycode) {
                    new Actions(driver).keyDown(keycode).perform();
```

```
}

public void keyUpNative(Keys keycode) {
        new Actions(driver).keyUp(keycode).perform();
}

public void keyPressNative(String keycode) {
        new Actions(driver).click().perform();
}


        public void waitForElementPresent(By by) {
                for(int i=0; i<60; i++) {
                if (isElementPresent(by)) {
                break;
                } else {
                try {
                driver.wait(1000);
                } catch (InterruptedException e) {
                e.printStackTrace();
                                }
                        }
                }
                }


        public void clickAndWaitForElementPresent(By by, By waitElement) {
                click(by);
                waitForElementPresent(waitElement);
                }


        public Boolean VeryTitle(String exception,String actual){
                if(exception.equals(actual)) return true;
                else return false;
        }
}
```

PS：有什么建议，欢迎评论，一起交流！

1.17 selenium webdriver学习（十七）----------把selenium项目同步到本地eclipse

发表时间: 2012-05-11

这里主要是想把selenium的整个项目同步到eclipse的经历分享一下。虽然有时候想想没有必要，因为你下载的包里本身就包含了源代码，但是我就是这样做了。

selenium项目地址：（只读）

方法一、直接使用 TortoiseSVN

1、安装TortoiseSVN。下载地址：http://tortoisesvn.net/downloads

2、在eclipse的Workspace目录下直接新建一个文件夹，右击文件夹 -> TortoiseSVN -> Export... ,在打开的弹层"URL of repository："下面的输入框输入**http**://selenium.googlecode.com/svn/trunk/,点击OK。

3、把同步下来文件夹做为一个项目导入eclipse。（File>>Import>>General>>Existing Projects into Workspace 然后选择自己的项目）

这样我们就可以在你自己的eclipse里看到selenium整个项目了。如下图，

selenium

方法二、使用Subclipse

Subclipse是一个eclipse插件。(包括 Eclipse PDT和 Zend Studio for Eclipse).它能在eclipse里面安装和更新。

subclipse安装步骤(比较懒，直接复制官网的)：

- Go to Help | Software Updates | Find and Install...
- Choose Search new features to install, Next.
- Choose New Remote Site, enter name "Subclipse" and URL "http://subclipse.tigris.org/update_1.8.x".
- Select the created "Subclipse" site and click Next.
- Follow the instructions and restart the workbench.

安装好了之后，重启eclipse，切换eclipse到SVN资源库

- 在eclipse工具栏，Window | open perspective | Other...
- 选择 "SVN 资源库研究，OK
- 在进入"SVN资源库"后，右击—>新建 - >资源库位置...
- 在弹出的层中输入 **http**://selenium.googlecode.com/svn/trunk/，Finish.
- 这样会在左边的面板出现这个添加的库，直接右击 - >检出为．．．
- 在弹出的框中随便输入项目名，Finish。

ok了，得到的项目和上图一样。

PS：有时候导出的项目会有报错，主要是第三方包有更新，看一下Path下的jar文件是不是有missing的情况。

1.18 "WebDriverException: Cannot find firefox binary in PATH."的解决方法

发表时间: 2012-02-07 关键字: webdriver, firefox, java, path

最近在学习webdriver,顺便把遇到的问题记在这里，以便日后查阅，并分享给遇到相同问题的人。

问题：运行seleniumhq.org网站上的例子。

```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.support.ui.ExpectedCondition;
import org.openqa.selenium.support.ui.WebDriverWait;

public class Selenium2Example  {
    public static void main(String[] args) {
        // Create a new instance of the Firefox driver
        // Notice that the remainder of the code relies on the interface,
        // not the implementation.
        WebDriver driver = new FirefoxDriver();

        // And now use this to visit Google
        driver.get("http://www.google.com");
        // Alternatively the same thing can be done like this
        // driver.navigate().to("http://www.google.com");

        // Find the text input element by its name
        WebElement element = driver.findElement(By.name("q"));

        // Enter something to search for
        element.sendKeys("Cheese!");

        // Now submit the form. WebDriver will find the form for us from the element
        element.submit();
```

```
        // Check the title of the page
        System.out.println("Page title is: " + driver.getTitle());

        // Google's search is rendered dynamically with JavaScript.
        // Wait for the page to load, timeout after 10 seconds
        (new WebDriverWait(driver, 10)).until(new ExpectedCondition<Boolean>() {
            public Boolean apply(WebDriver d) {
                return d.getTitle().toLowerCase().startsWith("cheese!");
            }
        });

        // Should see: "cheese! - Google Search"
        System.out.println("Page title is: " + driver.getTitle());

        //Close the browser
        driver.quit();
    }
}
```

报如下错误

Exception in thread "main" org.openqa.selenium.WebDriverException: Cannot find firefox binary in PATH. Make sure firefox is installed. OS appears to be: XP
Build info: version: '2.18.0', revision: '15704', time: '2012-01-27 17:37:17'
System info: os.name: 'Windows XP', os.arch: 'x86', os.version: '5.1', java.version: '1.6.0_23'

看这个报错应该是firefox安装路径不是默认路径。

解决方法：方法1、最简单的重新安装firefox到默认路径。哈哈

方法2、直接用System.setPropert方法设置webdriver.firefox.bin的值，如

```
System.setProperty("webdriver.firefox.bin","D:\\Program Files\\Mozilla Firefox\\firefox.exe");
```

方法3、 用FirefoxBinary类和public FirefoxDriver(FirefoxBinary binary, FirefoxProfile profile)这个构造方法，直接上代码

```
File pathToFirefoxBinary = new File("D:\\Program Files\\Mozilla Firefox\\firefox.exe");
FirefoxBinary firefoxbin = new FirefoxBinary(pathToFirefoxBinary);
WebDriver driver = new FirefoxDriver(firefoxbin,null);//这里使用这个构造方法。
```

应该还可以在环境变量里面设置firefox的路径也可以，有兴趣的可以试一下。

注：有人可能会不知道webdriver.firefox.bin，可以看一下源码，其中

org.openqa.selenium.firefox.internal.Executable.locateFirefoxBinaryFromSystemProperty()

方法第一句

```
String binaryName = System.getProperty("webdriver.firefox.bin");
```

说明默认的时候取的就是这个值，重新设置一下。