**Selenium使用介绍**

Selenium  
严格说来，Selenium是一套完整的Web应用程序**[测试](javascript:;" \t "_self)**系统，它包含了测试的录制（Selenium IDE）、编写及运行（Selenium Remote Control）和测试的并行处理（Selenium Grid）。Selenium的核心Selenium Core基于JsUnit，完全由JavaScript编写，因此可运行于任何支持JavaScript的浏览器上。Selenium Core由一种指定格式的HTML文件驱动，在一定程度上增强了测试套件（**[Test](javascript:;" \t "_self)** Suite）的可读性。Selenium Remote Control允许测试人员使用常见的语言（自然包括C#等.NET语言）编写测试代码，并支持不同**[操作系统](javascript:;" \t "_self)**下的各种主流浏览器。Selenium Grid的作用是将测试分发至多台机器，这样便可大大加快测试速度。与WatiN相同，Selenium也是一款同样使用Apache License 2.0协议发布的开源框架。  
  
  
  
Selenium IDE  
  
       Selenium IDE是基于FIREFOX浏览器的一个插件，提供GUI界面来运行Selenium测试。Selenium IDE提供脚本录制功能，可以将用户在浏览器中执行的操作记录下来，生成各种形式的脚本，可以将这些脚本保存供以后使用。  
  
       安装Selenium IDE的步骤如下：  
  
    \* 从www.openqa.org/selenium-ide/download.action下载Selenimu IDE（这是一个XPI后缀的文件)。  
    \* 启动FIREFOX浏览器，打开刚才下载的文件。  
    \* 重启FIREFOX浏览器，在工具菜条下应该就可以看到Selenium IDE菜单项  
  
  
  
---------------------------------------------------------------------------------------------------------  
  
一 Selenium-RC 是 selenium-remote control 缩写，是使用具体的语言来编写测试类。  
  
二 准备**[工作](javascript:;" \t "_self)**: 1,下载 selenium 了，到 http://www.openqa.org/selenium/ 下载就可以了，记得选择selenium-rc 的版本  
             2, [**学习**](javascript:;)一下xpath 的知识。有个教程：http://www.zvon.org/xxl/XPathTutorial/General\_chi/examples.html  
             3, 安装 jdk1.5  
  
  
三 selenium-rc 一些使用方法  
  
    1，解压selenium-rc压缩包  
  
    2，启动服务器   
        Selenium Server是用**[JAVA](javascript:;" \t "_self)**实现的，相应的库文件在HOME/server/selenium-server.jar。运行如下代码从命令行启动：  
       java 代码 : java -jar selunium-server.jar  
  
  
四   编写测试用例  
  
    需要的JAR: selenium-java-client-driver.jar;junit  
  
  
    编写一个JUNIT的**[单元测试](javascript:;" \t "_self)**，要构建一个Selenium，包括如下步骤：  
  
    \* 构建一个Selenium实例  
    \* 启动Selenium实例  
    \* 执行Selenium命令，并验证结果。  
    \* 关闭Selenium实例  
  
  
如下是一个示例，用来测试http://www.google.com/，查找selenium，期望结果中包含"OpenQA: Selenium"  
   1. package com.thoughtworks.selenium;       
   2.    
   3. import junit.framework.\*;    
   4.    
   5. import org.openqa.selenium.server.\*;    
   6.    
   7. public class GoogleTest extends TestCase    
   8. {    
   9.    private Selenium selenium;    
  10.    
  11.    public void setUp() throws Exception {    
  12.         String url = "http://www.google.com";    
  13.        selenium = new DefaultSelenium("localhost", SeleniumServer.getDefaultPort(), "\*firefox", url);    
  14.        selenium.start();    
  15.     }    
  16.        
  17.    protected void tearDown() throws Exception {    
  18.        selenium.stop();    
  19.    }    
  20.        
  21.    public void testGoogleTestSearch() throws Throwable {    
  22.         selenium.open("/intl/zh-CN/");    
  23.         selenium.type("q", "selenium");    
  24.         selenium.click("btnG");    
  25.         selenium.waitForPageToLoad("30000");    
  26.         assertEquals("selenium - [**Google**](javascript:;) 搜索", selenium.getTitle());  
  27.         
  28.     }    
  29.         
  30. }     
  
五 多环境测试  
  
package test;  
  
import org.junit.Test;  
  
import junit.framework.TestCase;  
  
import com.thoughtworks.selenium.DefaultSelenium;  
import com.thoughtworks.selenium.Selenium;  
/\*\*  
\* 多浏览器测试  
\* @author hgz  
\*  
\*/  
public class NewTest extends TestCase {  
@Test  
public void testNew() throws Exception {  
Selenium selenium = new DefaultSelenium("localhost", 4444, "\*chrome",  
"http://www.google.cn");  
script(selenium);  
}  
  
@Test  
public void testNew2() throws Exception {  
Selenium selenium = new DefaultSelenium("localhost", 4444, "\*iehta",  
"http://www.google.cn");  
script(selenium);  
}  
  
private void script(Selenium selenium) throws Exception {  
try {  
selenium.start();  
selenium.open("http://www.google.cn/");//调用 selenium.open 方法，浏览器会打开相应的页面  
selenium.type("q", "selenium");//使用 type 方法来给输入框输入文字  
selenium.click("btnG");  
selenium.waitForPageToLoad("30000");//等待页面载入  
assertEquals("selenium - Google 搜索", selenium.getTitle());//看看新的页面标题是不是我们想要的。  
} catch (Exception e) {  
throw e;  
} finally {  
selenium.stop();  
}  
}  
}  
  
  
六 如何选取元素  
  
   selenium提供如下强大的定位元素的方法。  
  
    \* id=id  
    \* name=name  
    \* dom=javascriptExpression  
    \* xpath=xpathExpression  
    \* link=textPattern  
    \* css=cssSelectorSyntax  
  
  
  
1 通过ID,name选择元素  ：  如 selenium.type("id=q","百度"); selenium.type("name=search","百度")  
2 link= 根据链接文字来操作:如 selenium.click("link=个人资料");  
3 根据XPath来选择元素  : XPath Checker  
  
    \* xpath=//img[@alt='The image alt text']  
    \* xpath=//table[@id='table1']//tr[4]/td[2]  
    \* xpath=//a[contains(@href,'#id1')]  
    \* xpath=//a[contains(@href,'#id1')]/@class  
    \* xpath=(//table[@class='stylee'])//th[text()='theHeaderText']/../td  
    \* xpath=//input[@name='name2' and @value='yes']  
    \* xpath=//\*[text()="right"]  
      如: selenium.type("xpath=//input[@name='user.email']", "xxx@123.com"); // 向input中type为text的栏位键入信息  
          selenium.check("xpath=//input[(@name='user.sex')and(@value='男')]");// 向input中type为radiod的 选取  
  
4 dom选择  
  
    \* dom=document.forms['myForm'].myDropdown  
    \* dom=document.images[56]  
    \* dom=function foo() { return document.links[1]; }; foo();  
  
5 css选择器  
  
这个不常用，它可以支持css2, css3选择器  
  
    \* css=a[href="#id3"]  
    \* css=span#firstChild + span  
  
  
  
七 使用selenium 这个对象来进行测试  
1  获取标 : assertEquals("Insert title here", selenium.getTitle());  
2  判断页面是否存在一个user.email元素  :assertTrue(selenium.isElementPresent("xpath=//input[@name='user.email']"));  
3  得到文本框里的文字:   assertEquals(selenium.getValue("xpath=//input[@name='user.username']"),"xxxaas");  
4  测试check box  :    assertTrue(selenium.isChecked("xpath=//input[(@name='user.sex')and(@value='男')]"));  
5  点击提交按钮   : selenium.click("xpath=//input[@type='button']");  
6  等待页面载入   : selenium.waitForPageToLoad("2000");  
7  验证指定文本出现在提交给用户的页面上: assertTrue(selenium.isTextPresent("验证码输入有误，请核实后再输入"));  
8  判断下拉框里选择了哪个选项 :assertEquals(selenium.getSelectedIndex("xpath=//SELECT[@name='HATIMING']"), "1");   
  
9  如何测试一些错误消息的显示？ assertTrue(selenium.getBodyText().indexOf("错误消息")>=0);   
   getBodyText 返回的时浏览器页面上的文字，不回包含html 代码的，如果要显示html 代码，用下面这个:selenium.getHtmlSource();   
  
八   Firefox 的插件  
     1 XPath Checker ：可以用这个工具测试简化我们的xpath表达式  
     2 Firebug  
     3 Selenium IDE  
     4 Execute JS