Web自动化测试工具

--Selenium

一、介绍 Selenium WebDriver

Selenium WebDriver 是一种用于Web 应用程序的自动测试工具，它提供了一套友好的API，与Selenium RC 相比，Selenium WebDriver 的API更容易理解和使用，其可读性和可维护性也大大提高。

Selenium WebDriver 完全就是一套类库，不依赖于任何测试框架，不需要启动其他进程或安装其他程序，也不必像Selenium RC 那样需要先启动服务。

可以近似认为：

Selenium WebDriver = Selenium RC + WebDriver

一、介绍 Selenium WebDriver

什么是 WebDriver ？

WebDriver 针对各个浏览器而开发，取代了嵌入到被测Web应用中的JavaScript。与浏览器的紧密集成支持创建更高级的测试，避免了JavaScript安全模型导致的限制。

除了来自浏览器厂商的支持，WebDriver还利用操作系统级的调用模拟用户输入。WebDriver支持 Firefox (FirefoxDriver)、IE (InternetExplorerDriver)、Opera (OperaDriver) 和Chrome (ChromeDriver)。它还支持Android (AndroidDriver)和iPhone (IPhoneDriver) 的移动应用测试。它还包括一个基于HtmlUnit的无界面实现，称为HtmlUnitDriver。

WebDriver API可以通过Python、Ruby、Java和C#访问，支持开发人员使用他们偏爱的编程语言来创建测试。

一、介绍 Selenium WebDriver

WebDriver 设计思想

WebDriver开发团队将WebDriver的API定位为“基于对象的”。

接口被明确定义并努力坚持只包含一个角色或者责任，而不是将每一个可能的HTML标记模块化为单独的类，我们只有一个WebElement接口。通过这种方式，开发人员使用支持自动补全的IDE即可被提示下一步工作。

```
WebDriver driver = new FirefoxDriver();
driver.<user hits space>
```

一、介绍

Selenium WebDriver

WebDriver 与 RC的对比

早期Selenium API的方法：

type
typeKeys
typeKeysNative
keydown
keypress
keyup
keydownNative
keypressNative
keyupNative
attachFile

WebDriver 中的等价方法：

sendKeys

一、介绍 Selenium WebDriver

WebDriver项目与 RC项目合并

为何把两个项目合并？

部分原因是WebDriver解决了Selenium存在的缺点（比如，能够绕过JS沙箱。我们有出色的API）

部分原因是 Selenium解决了WebDriver存在的问题（例如支持广泛的浏览器）

部分原因是因为Selenium的主要贡献者和我都觉得合并项目是为用户 提供最优秀框架的最佳途径。

二、使用WebDriver

环境准备

官网：

java版本：2.48.2 2015-10-09

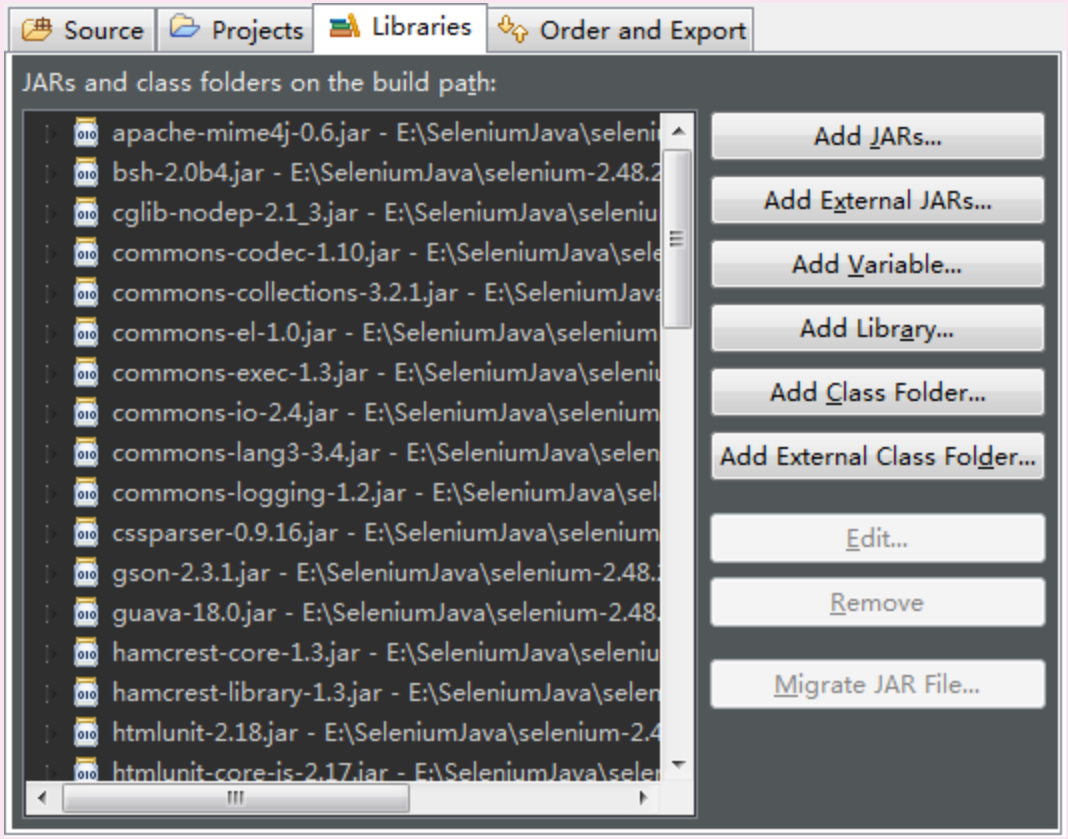
解压，得到以下文件夹：

libs文件夹：包含各种Java相关的基础框架。
CHANGELOG：记录Selenium的变更情况。
Selenium-java-2.48.2.jar：Selenium主要API文件，在进行自动化测试是主要就靠这个类库来实现。
Selenium-java-2.48.2-srcs.jar：Selenium部分源码。

二、使用WebDriver

环境准备

将Selenium-java-2.48.2.jar以及libs目录下的包导入到工程。
注意，环境变量或许要添加浏览器的安装路径。



二、使用WebDriver

第一个程序

```
# coding = utf-8
from selenium import webdriver

browser = webdriver.Firefox()
browser.get("http://www.baidu.com")

browser.find_element_by_id("kw").send_keys("selenium")
browser.find_element_by_id("su").click()
```

此处换成java，各个浏览器都调用下

二、使用WebDriver

元素定位

python 版本：2.48.0 2015-10-07

对象的定位是自动化测试的核心， 要想操作一个对象， 首先应该识别这个对象。

一个对象就是一个人一样，他会有各种的特征（属性），比如我们可以通过一个人的身份证号，姓名，或者他住在哪个街道、楼层、门牌找到这个人。

那么一个对象也有类似的属性，我们可以通过这个属性找到这对象。

二、使用WebDriver

元素定位

WebDriver 提供了一系列的对象定位方法，常用的有以下几种：

- id
- name
- class name
- link text
- partial link text
- tag name
- css selector
- xpath

```
<input id="query" class="" type="text"  plete="off"
maxlength="100" size="47" name="query">
```

二、使用WebDriver

元素定位(id)

```java
package example;

import org.openqa.selenium.*;
import org.openqa.selenium.WebDriver.*;
import org.openqa.selenium.firefox.*;
import org.openqa.selenium.chrome.*;

public class exampleClass {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        WebDriver firefox = new FirefoxDriver();

        Navigation fireNav = firefox.navigate();
        fireNav.to("http://www.sogou.com");

        WebElement sogouTextBox = firefox.findElement(By.id("query"));
        sogouTextBox.sendKeys("找到文本框");

        firefox.close();
        //WebDriver chrome = new ChromeDriver();
    }
}
```

二、使用WebDriver

元素定位(name)

```
public static void main(String[] args) {
    // TODO Auto-generated method stub
    WebDriver firefox = new FirefoxDriver();

    Navigation fireNav = firefox.navigate();
    fireNav.to("http://www.sogou.com");

    WebElement sogouTextBox = firefox.findElement(By.name("query"));
    sogouTextBox.sendKeys("selenium");

    firefox.close();
}
```

二、使用WebDriver

元素定位(class name)

```
firefox.findElement(By.className("s-logion")).click();
```

元素定位(link text)

```
firefox.findElement(By.linkText("登录")).click();
```

元素定位(partial link text)

```
firefox.findElement(By.partialLinkText("录")).click();
```

二、使用WebDriver

元素定位(tagname)

```
List<WebElement> tags = firefox.findElements(By.tagName("a"));
```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：https://d.book118.com/966140102154010135