在 AC 上开发 selenium 脚本

目 录

作者：柳胜

1. 开发 selenium Case

1.1 基于录制回放的 selenium case

录制一个简单的 web 计算器功能，export 到 junit 模式，保存为 selJava.java 文件，如下：

```
package com.example.tests;
import com.thoughtworks.selenium.*;
import java.util.regex.Pattern;
public class selJava extends SeleneseTestCase {
    public void setUp() throws Exception {
        setUp("http://change-this-to-the-site-you-are-testing/", "*chrome");
    }
    public void testSelJava() throws Exception {
        selenium.open("/calc.htm");
        selenium.click("//input[@name='alex' and @value='1']");
        selenium.click("//input[@name='alex' and @value='+']");
        selenium.click("//input[@name='alex' and @value='2']");
        selenium.click("//input[@name='alex' and @value='=']");
        verifyEquals("3", selenium.getValue("display"));
    }
}
```

以上 Selenium case 继承 SeleneseTestCase，SeleneseTestCase 的父类是 Junit TestCase，因此，在默认模式下，Selenium Case 实际上是以 Junit Runner 方式运行的。

Junit 本身是 java 单元测试框架，并不完全满足我们 selenium 的测试需求，比如对 case 之间的依赖关系及数据交互，web 测试抓图等等，junit 并不能胜任。

1. Selenium 代码从 Junit 模式转化到 AC 模式

可遵循以下步骤，将 junit 模式的 selenium 代码转化成在 AC 模式：

1. 在 java 环境中的 classPath 添加 ac_framework.jar
2. 在 case 文件头添加 import framework.JobDOM.ACSeleniumJob;

将 selJava 的父类改为 ACSeleniumJob，

public class selJava extends ACSelniumJob

3. 由于录制生成的 setup 函数调用了父类的 setup，封装了 selenium 初始化的操作。在这里，我们直接将 selenium 的初始化操作放到子类的 setup 函数中。
4. 增加函数 public void tearDown()，做收尾工作。

Ok，到这里，新的 selenium case 已经完成了（红色为修改处），如下：

```
package com.example.tests;
import com.thoughtworks.selenium.*;
import java.util.regex.Pattern;
import framework.JobDOM.ACSeleniumJob
public class selJava extends SeleneseTestCase {
    public void setUp() throws Exception {
        selenium = new DefaultSelenium("localhost", 4444, "*firefox C:\\Program Files\\Mozilla Firefox\\firefox.exe", " http://localhost/calc.htm ");
    }
    public void testSelJava() throws Exception {
        selenium.open("/calc.htm");
        selenium.click("//input[@name='alex' and @value='1']");
        selenium.click("//input[@name='alex' and @value='+']");
        selenium.click("//input[@name='alex' and @value='2']");
        selenium.click("//input[@name='alex' and @value='=']");
        verifyEquals("3", selenium.getValue("display"));
    }
    public void tearDown() throws Exception {
```

```
        selenium.stop();
    }
}
```

2. 将 Selenium case 添加到 TestJobFile 中，运行 AC，获得测试结果

TestJobFile 中添加 Selenium Job，按如下格式定义

```
<TestJob name="selenium_demo" description="Test calc"   depends="" driver_type="SELENIUM" >
        <TestData type="xml"   location="selenium\config.xml"/>
        <JobInput name="$MAIL_SUBJECT"/>
        <ClassPath location="selenium\selenium-java-client-driver_self_extended_oracle.jar"/>
        <ClassPath location="selenium\orajtst.jar"/>
        <ClassPath location=" selenium\qa.jar"/>
        <ClassPath location=" selenium\selJava.class"/>
        <SelTestCase path="selJava">
            <SelTest name=" testSelJava "/>
        </SelTestCase>
</TestJob>
```

运行 AC framework，即可执行 selenium Job，并获得测试报告

1.2 将启动 selenium server 集成到 TestJob 中

我们最常用的是 Selenium RC 模式，即先启动一个 selenium server，然后才能运行 selenium 脚本。

java 启动 selenium server 的命令行语句如下：

java –jar selenium-server.jar –port 4444 – proxyInjectionMode –log selServer.txt

如何把启动 selenium server 的 java 命令行也集成到 AC 中来呢？

这里要用到 AC 的 Ant Engine。

创建基于 Ant Engine 的 TestJob，TestJob 内容遵循 Ant 语法，如下:

```
<TestJob name="Ant_StartSelenium" description="selenium initialization"   depends="" driver_type="ANT" daemon="true">
        <java fork="true" spawn="true" jar="D:\selenium-server.jar">
            <arg line="-port 4444 "/>
```

```
            <arg line="-proxyInjectionMode"/>
            <arg line="-log sel.txt"/>
        </java>
    </TestJob>
```

启动 Selenium server 的 TestJob 可与 Selenium Test Job 做一个 dependence 的定义，保证每次运行 selenium 测试的时候，selenium server 是处于启动状态的

<TestJob name="Ant_StartSelenium" description="selenium initialization" depends="" **driver_type="ANT"** daemon="true">

……

</TestJob>

<TestJob name="selenium_demo" description="Test calc" **depends=" Ant_StartSelenium " driver_type="SELENIUM"** >

……

</TestJob>

1.3 框架提供的 selenium API

reportPass(String msg): 向 AC 汇报当前运行状态，为成功

reportFail(String msg)：向 AC 汇报当前运行状态，为失败

reportWarning(String msg)：向 AC 汇报当前运行状态，为警告

getDataProperty(String key)：获得测试数据

getEnvProperty(String key)：获得环境变量

getConfProperty(String key)：获得配置数据

getInputValue(String key)：从全局数据通道中获得输入数据

setOutputValue(String key,String value)： 向全局数据通道输出数据