参考： <http://www.cnblogs.com/hyddd/archive/2009/05/20/1473146.html>

**一.Selenium是什么？**

Selenium是ThroughtWorks公司一个强大的开源Web功能测试工具系列，本系列现在主要包括以下4款：

        1.Selenium Core：支持DHTML的测试案例（效果类似数据驱动测试），它是Selenium  IDE和Selenium  RC的引擎。

        2.Selenium IDE：FireFox的一个插件，支持脚本录制（实际上是将操作流程输出为一个脚本）。

        3.Selenium RC（Remote Control）：Selenium Remote Control。后续的系列文章我会主要针对Selenium RC展开介绍。

        4.Selenium Grid（分布式测试）：允许同时并行地、在不同的环境上运行多个测试任务，极大地加快Web应用的功能测试。

**二.选择合适的Selenium工具**

既然Selenium工具有4款这么多，那到底如何选择呢？？我从"[Selenium官网](http://wiki.openqa.org/pages/viewpage.action?pageId=763)"这里找了一个表：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Selenium IDE | Selenium Remote Control | Selenium Core | Selenium Core HTA |
| 浏览器支持 | 仅Firefox | 很多 | 所有 | 仅IE |
| 需要远程安装 | 否 | 否 | 是 | 否 |
| 支持 HTTPS/SSL | 是 | 是\* | 是 | 是 |
| 支持跨域 | 是 | 是\* | 否 | 是 |
| 需要Java | 否 | 是 | 否 | 否 |
| 将测试结果保存到磁盘 | 是 | 是 | 否 | 是 |
| 多语言支持 | 仅Selenese | 很多 | 仅Selenese | 仅Selenese |

这里没有介绍Selenium Grid，但介绍了另外一个Selenium Core HTA，Selenium Core HTA其实是Selenium Core的额外模式，你只要Selenium Core配置稍加修改，即为HTA模式，Selenium Core HTA可以在IE最高安全等级（特权）下工作，这意味着它仅能在IE下工作，由于限制较大，下面将排除对Selenium Core HTA的讨论。

**1.浏览器支持**：

    (1).Selenium IDE仅可以在Firefox中工作。

    (2).Selenium Remote Control支持很多浏览器，包括最常用的：firefox，ie，safari等N款浏览器。

    (3).Selenium Core支持的浏览器是最广的，这点和它的实现有关。作为IDE和RC的引擎，Selenium Core几乎可以在任何浏览器中工作。

**2.需要远程安装**：是否需要在被测网站的服务端安装？

    这里只有Selenium Core需要，这是出于同源策略的原因。这也是Selenium Core一个很大的限制，试问，如果你要测试Google.com，还得在google的服务器上装一个Selenium Core，那是多搞笑的一件事。

    而Selenium IDE和Selenium Core HTA不会被同源策略所限制，因为他们对浏览器扩展了。

     Selenium RC提供一个代码服务器来保证Selenium JS文件看似来自相同的远程服务器，从而符合同源策略；代理服务器欺骗浏览器，让它认为这里的确有像<http://www.google.com/selenium/>这样的目录。

**3.支持HTTPS/SSL**：

    这里不说了，都支持。Selenium RC在“是”后面加\*因为它是在最近版本支持的，仅此而已。

**4.需要Java**：准确的说是需要JRE

    这项只有Selenium RC需要，上面2中所说的“代理服务器”是一个Java程序，需在跑测试案例前启动。

**5.将测试结果保存到磁盘**

    只有Selenium Cor（就是一大堆提供支持的js函数库）e不能将任何测试结果写到磁盘上（因为它是用javascript写的，它不允许向磁盘写数据），其解决方案是当然你可以将测试结果发送到另外一台服务器保存。这也是Selenium Core的一大限制。

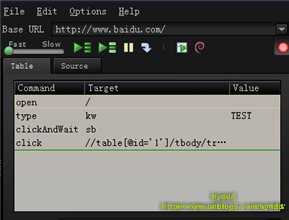
**6.多语言支持**

    (1).Selenium IDE仅支持Selenium语言。

    (2).Selenium RC支持很多语言，如：C#，Java，Python，Ruby等。

    (3).Selenium Core也是仅支持Selenium语言。

Selenium语言的测试案例如下：

[[](http://images.cnblogs.com/cnblogs_com/hyddd/WindowsLiveWriter/Selenium1Selenium_13256/seleniumLanguage_2.jpg)](http://images.cnblogs.com/cnblogs_com/hyddd/WindowsLiveWriter/Selenium1Selenium_13256/seleniumLanguage_2.jpg)

它的优点是：简单，用（Command，Target，Value）三种元素组成一个行为，并且有辅助录制脚本工具（如：Firefox IDE,Selenium Core等），不懂程序的测试人员都可以轻松地编写测试案例。

它的缺点是：Selenese有一些严格的限制：它没有条件（没有"if"表达式），并且它没有循环（没有"For"表达式）。这会使编写复杂的测试变得困难甚至不可能。

    OK，现在我们来研究下到底该使用哪款工具开展测试！

    (1).Selenium IDE支持并且只支持Firefox浏览器，支持的浏览器太少，而依附于Firefox也不便于日后开展自动化测试，但是，它的录制快捷好用！并且有代码转换功能，可以把Selenium语言测试案例转为C#,Java等语言的测试案例，我建议使用Selenium IDE + FireBug进行测试案例的编写，然后转为其他语言的测试案例后，再调用Selenium RC运行测试案例。

    (2).Selenium Core，它的优点是编写测试案例简单，并且支持绝大多数的浏览器，但缺点也同样明显，Selenium Core需要远程安装（在测试的服务器端安装），Selenese语言也限制了复杂案例的可能性，并且没有良好的外部扩展，这是些都会是致命的问题。因为一款测试工具不可能100%满足你测试需求的，当它不能满足你测试需求时候，它必须有一个扩展机制可以让你可以使用其他的方式满足你需求，否则这款测试软件即使功能强大，也请三思慎用，否则当投入大量资源后才发现某些问题不能解决，那时候已经晚了，这是我的切身体会。

    (3).Selenium RC是我推荐使用的工具，它支持很多浏览器，可以使用C#，Java等语言编写测试案例，易于维护，同时提供了很好的扩展性，所以后续的文档我会以Selenium RC作为默认的测试工具。