

Selenium入门

欢迎阅读Selenium入门讲义，本讲义将会重点介绍Selenium的入门知识以及Selenium的前置知识。

##自动化测试的基础

在Selenium的课程以前，我们先回顾一下软件测试的基本原理，为我们进一步完成Selenium自动化测试做准备工作。

软件测试的发展

G.J.Myers在其经典的著作《软件测试艺术》（The Art of Software Testing）一书中，给出了测试的定义：

“程序测试是为了发现错误而执行的过程。”

这个概念产生于30年前，对软件测试的认识还非常有局限性，当然也是因为受瀑布开发模型的影响，认为软件测试是编程之后的一个阶段。只有等待代码开发出来以后，通过执行程序，像用户那样操作软件去发现问题。

上述也就是动态测试。

随后延伸到静态测试，也就是在对于写代码之前的文档、以及编码过程中的代码进行代码扫描、评审等。

当前，软件测试贯穿到整个软件开发生命周期的全过程中，不再停留在编程之后的某个阶段，尤其是敏捷开发开始广泛的应用于互联网行业以后，敏捷测试就把软件测试解释为对软件产品质量的持续评估。在敏捷方法中，持续测试被提倡。当前的持续测试的实施，主要依托于持续集成。

持续集成和持续测试的产生和发展，对自动化测试产生了比较高的要求。此外随着大数据和机器学习的发展，对自动化测试的需求越来越迫切。

简单的描述一下持续集成，Continuous Integration，简称CI。是一种软件工程实践：项目需要经常性的集成，进行代码编译，版本构建，自动化的部署以及部署后的自动化测试。也是云开发测试模式的一种。

持续集成构成了持续交付的基础。

自动化测试：以人为驱动测试行为转化为机器执行的一种过程

软件测试的维度

- 从软件测试的过程看：单元测试、集成测试、确认测试、系统测试、验收测试
- 从软件测试的方法看：白盒测试、黑盒测试、灰盒测试
- 从软件测试的测试点看：功能测试、性能测试、兼容性测试、安全测试、探索性测试
- 从软件测试的技术程度看：手工测试、自动化测试、测试开发

自动化测试的场景

自动化测试应用的场景：

- 功能测试的自动化：测试界面显示、测试业务逻辑、测试数据导入导出、测试接口功能
- 性能测试的自动化：测试压力和负载、测试接口的性能等

什么样的项目适合自动化测试

1. 任务测试明确，不会频繁变动
2. 每日构建后的测试验证
3. 比较频繁的回归测试
4. 软件系统界面稳定，变动少
5. 需要在多平台上运行的相同测试案例、组合遍历型的测试、大量的重复任务
6. 软件维护周期长
7. 项目进度压力不太大
8. 被测软件系统开发比较规范，能够保证系统的可测试性
9. 具备大量的自动化测试平台
10. 测试人员具备较强的编程能力

自动化测试工具介绍

- HP UFT (QTP)
- Selenium

- HP LoadRunner
- Robot Framework
- Watir (webapplication test in ruby)
- SoapUI
- MonkeyRunner

Web测试技术

Selenium是一款出色的Web自动测试工具，那么在使用之前，Web测试的基本技术也是必须需要了解的。

####Web前端技术

基础技术	高阶技术
Http协议：报文、请求、Cookie等	Ajax
Html和CSS	Html5+CSS3
JavaScript语言和框架（其中jQuery是基础框架）	Bootstrap
JSON和XML	AngularJS

推荐使用的前端工具：Firebug。FireFox火狐浏览器所自带的一个前端插件。

<https://addons.mozilla.org/en-US/firefox/addon/firebug/>

<http://getfirebug.com>

什么是Selenium

Selenium

Selenium is a suite of tools to automate web browsers across many platforms.

硒，`/s'linim/`

Selenium是开源的自动化测试工具，它主要是用于Web 应用程序的自动化测试，不只局限于此，同时支持所有基于web 的管理任务自动化。

Selenium 是用于测试 Web 应用程序用户界面 (UI) 的常用框架。它是一款用于运行端到端功能测试的超强工具。您可以使用多个编程语言编写测试，并且 Selenium 能够在一个或多个浏览器中执行这些测试。

Selenium经历了两个版本，Selenium1.0 和Selenium2.0，Selenium 也不是简单一个工具，而是由几个工具组成，每个工具都有其特点和应用场景。

Selenium是在2004年由一个叫Jason Huggins的聪明的年轻人开发的，当时身处ThoughtWorks的他，为了不想让自己的时间浪费在无聊的重复性工作中，所以开发一个JS的类库来驱动浏览器页面的行为；这个js类库就是selenium core，同时也是seleniumRC、Selenium IDE的核心组件。这就是Selenium1.0的产生史。

在2006年，在google的一位有胆识的年轻人Simon Stewart发起了一个叫WebDriver的项目；因为长期以来google一直是selenium的重度用户，但却总是被限制在有限的操控范围内，所以Simon希望能通过浏览器、操作系统的底层方法等一些手段来直接操作浏览器；这样就避免了在JavaScript的沙箱环境里存在的那些限制了，webdriver项目就此诞生！

在2008年，Selenium和webdriver结合了，从此永结良缘，福泽我们广大的测试工作者。这期间谁追的谁都不重要了，重要的是他们合并的原因正如webdriver的作者所说：

- 部分原因是selenium补充了webdriver的不足
- 部分原因是webdriver补充了selenium的不足
- 部分原因是它们合体后能给用户提供一个更好的自动化测试框架

所以管它呢，现在大家有一个比较好用的selenium就行了！

Selenium 1 和 WebDriver 合并成一款性能更佳的产品 Selenium 2（或 Selenium WebDriver），该款产品发行于 2011 年。Selenium 2 具有来自 WebDriver 的清晰面向对象 API，并能以最佳的方式与浏览器进行交互。Selenium 2 不使用 JavaScript 沙盒，它支持多种浏览器和多语言绑定。

为什么选择Selenium

Selenium特点：

- 开源，免费
- 多浏览器支持：Firefox、Chrome、IE、Opera、Safari
- 多平台支持：Linux、windows、Mac
- 多语言支持：Java、Python、ruby、PHP、C#、JavaScript
- 对web 页面有良好的支持
- 简单（API 简单）、灵活（用开发语言驱动）
- 支持分布式测试用例执行

Selenium的家谱

Selenium到目前为止已经经历了两个版本：Selenium 1.0和Selenium2.0.

Selenium官方网站：<http://docs.seleniumhq.org>

Selenium github：<https://github.com/SeleniumHQ/selenium>

Selenium1.0：包括 Selenium IDE，Selenium Grid和Selenium RC（Selenium Remote Control）

Selenium IDE 是嵌入到Firefox 浏览器中的一个插件，实现简单的浏览器操作的录制与回放功能。

那么什么情况下用到它呢？

快速的创建bug 重现脚本，在测试人员的测试过程中，发现了bug 之后可以通过IDE 将重现的步骤录制下来，以帮助开发人员更容易的重现bug。

IDE 录制的脚本可以转换成多种语言，从而帮助我们快速的开发脚本。

Selenium IDE 下载地址：<https://addons.mozilla.org/en-US/firefox/addon/selenium-ide/>

Selenium Grid 是一种自动化的测试辅助工具，Grid 通过利用现有的计算机基础设施，能加快WebApp的功能测试。利用Grid，可以很方便地同时多台机器上和异构环境中并行运行多个测试事例。其特点为：

- 并行执行
- 通过一个主机统一控制用例在不同环境、不同浏览器下运行。
- 灵活添加变动测试机

Selenium RC 是selenium 家族的核心工具，Selenium RC 支持多种不同的语言编写自动化测试脚本，通过selenium RC 的服务器作为代理服务器去访问应用从而达到测试的目的。

selenium RC 使用分Client Libraries 和Selenium Server。

- Client Libraries 库主要用于编写测试脚本，用来控制selenium Server 的库。
- Selenium Server 负责控制浏览器行为，总的来说，Selenium Server 主要包括3 个部分：Launcher、Http Proxy、Core。

Selenium Core 是被Selenium Server 嵌入到浏览器页面中的。Selenium Core就是一堆JavaScript函数的集合，就是通过这些JavaScript函数，我们才可以实现用程序对浏览器进行操作。

Launcher 用于启动浏览器，把Selenium Core 加载到浏览器页面当中，并把浏览器的代理设置为Selenium Server 的Http Proxy。

Selenium2.0：包括了WebDriver。其实在Selenium2.0中主推的就是WebDriver这个模块。事实上WebDriver是Selenium RC的替代品，因为Selenium需要保留向下兼容性的原因，Selenium RC才没有被彻底的抛弃，如果使用Selenium开发一个新的自动化测试项目，那么我们强烈推荐使用Selenium2.0的WebDriver进行编码。

Selenium2.0 = Selenium1.0 + WebDriver

Selenium RC的实质是在浏览器中运行一个JavaScript应用程序，使用每个浏览器内置的JavaScript解释器来解释和执行Selenium的命令集合。

Selenium WebDriver与上述的Selenium RC是不同的。WebDriver针对不同的浏览器进行开发，取代了RC那样嵌入到被测试的WebApp程序中的JavaScript，使用了与浏览器更加集成的方式进行测试，避免了RC使用JavaScript而可能遇到的安全模型限制导致的问题等。

在此我们也更加推荐读者使用Selenium2.0的WebDriver开始学习。

编程语言的选择

Selenium1.0代表的是经典，Selenium2.0代表的是未来。

我们在此重点关注Selenium2.0的WebDriver。而WebDriver面临的第一个选择就是编程语言。WebDriver支持众多的编程语言，例如，C#、

Ruby、Python、Java、Javascript、Objective-C、PHP。我们在使用之前必须要确定一种。

编程语言的选择，我们选了Java。而没有选择更加流行的C#或者Python，主要出于以下几个原因。

Java 语言特性

- 面向对象
 - Java语言提供类、接口和继承等原语，为了简单起见，只支持类之间的单继承，但支持接口之间的多继承，并支持类与接口之间的实现机制（关键字为implements）。Java语言全面支持动态绑定，而C++语言只对虚函数使用动态绑定。总之，Java语言是一个纯的面向对象程序设计语言。
- 平台无关
 - Java程序（后缀为java的文件）在Java平台上被编译为体系结构中立的字节码格式（后缀为class的文件），然后可以在实现这个Java平台的任何系统中运行。这种途径适合于异构的网络环境和软件的分发。
 - 支持Mac，Windows，Linux，UNIX
- 多线程
 - 在Java语言中，线程是一种特殊的对象，它必须由Thread类或其子（孙）类来创建。通常有两种方法来创建线程：其一，使用型构为Thread(Runnable)的构造子将一个实现了Runnable接口的对象包装成一个线程，其二，从Thread类派生出子类并重写run方法，使用该子类创建的对象即为线程。值得注意的是Thread类已经实现了Runnable接口，因此，任何一个线程均有它的run方法，而run方法中包含了线程所要运行的代码。线程的活动由一组方法来控制。Java语言支持多个线程的同时执行，并提供多线程之间的同步机制（关键字为synchronized）。
- 安全性
 - Java通常被用在网络环境中，为此，Java提供了一个安全机制以防恶意代码的攻击。除了Java语言具有的许多安全特性以外，Java对通过网络下载的类型具有一个安全防范机制（类ClassLoader），如分配不同的名字空间以防替代本地的同名类、字节代码检查，并提供安全管理机制（类SecurityManager）让Java应用设置安全哨兵。