**初识Selenium**

参考：<http://www.jianshu.com/p/71c3d352b481>

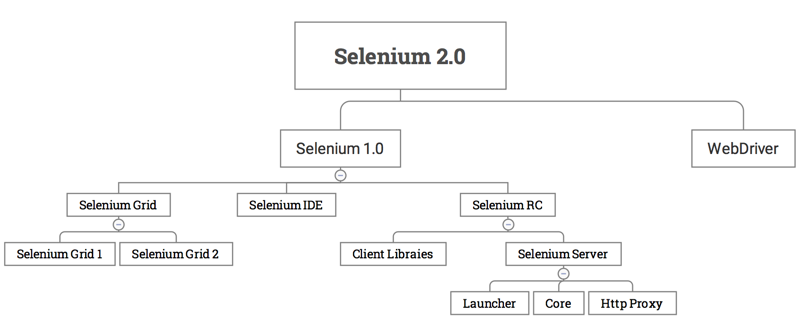
[](http://www.jianshu.com/u/06c5666f5bad)

**什么是Selenlium**

　　Selenium 是用于Web应用程序自动化测试的开源工具。通过Selenium，你可以模拟用户在浏览器中的操作。  
　　Selenium 它是一款用于运行端到端（endpoint2endpoint）功能测试的工具，你可以使用你熟悉的编程语言来写测试脚本，并且Selenium能够在一个或者多个浏览器中执行这些测试脚本。

**Selenium 家族介绍**

　　Selenium 经历了Selenium 1.0 到 Selenium 2.0 两个大版本，如今Selenium 3.0 也已经被提上了议程。  
　　Selenium1.0 源于ThoughtWorks公司，后跟google 的WebDriver联姻从而才有今天的Selenium 2.0（也有叫Selenium WebDriver）。下面通过一张图看看Selenium 的整个家族：

Selenium 家族.png

　　从上图看出，Selenium 不仅仅是一个测试工具，而是由多个工具组成，每个工具有其特点和其应用场景。

**Selenium 1.0**

　　Selenium 1.0 包括了 Selenium IDE, Selenium Grid和Selenium RC(Selenium Remote Control)。  
　　Selenium IDE 是一个Firefox 浏览器的一个插件，可以实现对浏览器的简单操作的录制与回放功能。一般用于测试人员在测试过程中发现bug，通过Selenium IDE 将重测试步骤录制下来，快速创建了bug 重现脚本，从而供开发人员容易重现bug。IDE 除了录制回放外还可以导出录制的脚本，并导出成多种语言、不同版本的Selenium 脚本，供测试人员快速开发脚本。  
　　Selenium Grid 是一个分布式执行 Selenium 脚本工具，现在大版本也到Selenium Grid 2。通过Selenium Grid 可以快速的在多台机器多个浏览器中并行执行多个测试用例，其大大提高了执行测试的效率，并通过一个hub节点管理多个不同系统不同浏览器的node节点，使脚本在不同的环境下被执行，可以测试兼容性，同样也便于添加变动的测试机器。  
　　Selenium RC 是Selenium 1.0 的核心，Selenium RC 分Clent Libraies 和 Selenium Server。  
　　Client Libraies 用于编写测试脚本，支持多种不同的主流开发语言，用来控制Selenium Server.  
　　Selenium Server 则负责控制浏览器的行为，而Selenium Server 又由三部分组成：

* Selenium Core：被Selenium Server 嵌入到浏览器页面中，是一堆的JavaScript 函数的集合，通过这些JavaScript函数来实现对浏览器的操作。
* 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Selenium Server 的Http Proxy。
* Http Proxy：Selenium Server的http代理。

**Selenium 2.0**

　　Selenium 2.0 包括了Selenium 1.0 和 WebDriver，然而WebDriver 其实是Selenium RC的替代品，比起Selenium RC ，WebDriver 提供一套更加面向对象，更易于使用的API，同时更容易操作浏览器和页面元素。所以现在我们学习Senlium 时更加推荐学习WebDriver。

**Selenium 的工作原理**

　　了解了Selenium 家族后，我们来聊聊Selenium RC 和 WebDriver 的工作原理。

**Selenium RC**

　　Selenium RC 使用的是javascript注入的方式跟浏览器打交道。这样 Selenium RC 需要启动一个Server，然后将操作页面元素的API 转成javascript脚本，再把这段脚本注入到浏览器中去执行。而通过这种javascript注入的方式一来太依赖翻译成javascript质量的好坏，二来javascript存在同源问题。这使测试变得不那么容易。

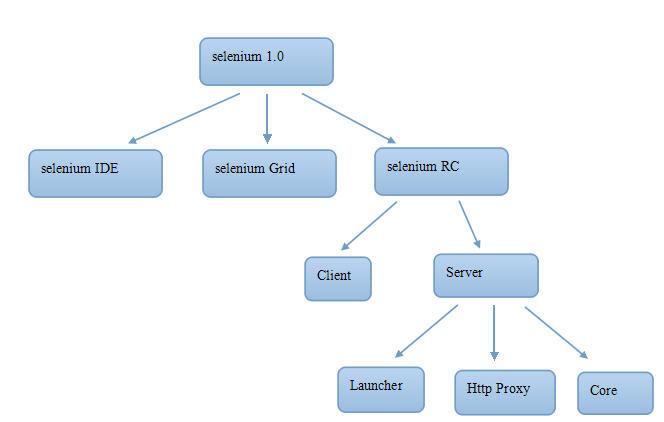
**WebDriver**

　　与Selenium RC 不同的是Selenium WebDriver 针对不同的浏览器进行独立开发Driver，利用浏览器的原生API去直接操作浏览器和页面元素，这样大大提高了测试的稳定性和速度。当然因为不同的浏览器对Web元素操作和呈现多多少少会存在一些差异，这也就造成现在不同的浏览器需要有对应不同的Driver（ChromeDriver ,IEDriver等等。。。）。

参考：<http://www.jianshu.com/p/b3ca4c5dccfe>

# 自动化测试---工具selenium介绍

首选你在开始selenium之前，需要花一到两个月时间去学一门语言，这里是根据没有语言基础的同学而定的。我推荐ruby ,python ,java 任意一门语言来进行学习。  
　　当然，已经如果有很好的语言基础略过这个环节，或者你的丰富的java编程能力，那么学习python 可能只需要几天时间或更短。  
　　假如，你已经搞定了一门语言的基础，接下来你需要先了解selenium ，selenium 并不是单纯的一个工具，他是一组工具的集合，而且，他还有1.0与2.0之分，当然3.0也已经到来。  
　　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 IDE  
　　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（这就说明ide录制的结果是产生一个重现bug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，关于这个功能后而用到时再详细介绍。

selenium Grid  
　　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Web-app的功能测试。利用Grid，可以很方便地同时在多台机器上和异构环境中（分布式测试辅助）并行运行多个测试实例。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  
　　selenium RC（Remote Control） 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。其中Selenium Core是被Selenium Server嵌入到浏览器页面中的。其实Selenium Core就是一堆JS函数的集合，就是通过这些JS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selnium 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。

selenium 2.0  
　　搞清了selenium 1.0 的家族关系，selenium 2.0 是把WebDriver 加入到了这个家族中；简单用公式表示为：  
　　selenium 2.0 = selenium 1.0 + WebDriver  
　　需要强调的是，在selenium 2.0 中主推的是WebDriver ，WebDriver 是selenium RC 的替代品，因为 selenium 为了向下兼容性，所以selenium RC 并没有彻底抛弃，如果你使用selenium开发一个新自动化测试项目，强列推荐使用WebDriver 。那么selenium RC 与webdriver 主要有什么区别呢？  
　　selenium RC 在浏览器中运行JavaScript应用，使用浏览器内置的JavaScript 翻译器来翻译和执行selenese命令（selenese 是selenium命令集合）。  
　　WebDriver通过原生浏览器支持或者浏览器扩展直接控制浏览器。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。  
　　如果是新项目直接学习webdriver 就OK了，RC是过时技术。

**selenium学习路线**

　　配置你的测试环境，针对你所学习语言，来配置你相应的selenium 测试环境。selenium 好比定义的语义---“问好”，假如你使用的是中文，为了表述问好，你的写法是“你好”，假如你使用的是英语，你的写法是“hello”。 所以，同样有语义在不同的语言下会有不同的写法（语法）。  
　　接着你需要熟悉webdriver API ，API就是selenium 所定义的方法，用于定位，操作页面上的各种元素。  
　　先学习元素的定位，selenium 提供了id、name、class name、 tag name、link text、partial link text、 xpath、css、等定位方法。xpath和css 功能强大语法稍微复杂，在这其间你可能还需要了解更多的前端知识。xml ,javascript 等。  
　　定位元素的目的是为了操作元素，接着就要学习各种元素的操作，输入框，下拉框，按钮点击，文件上传、下载，分页，对话框，警告框...等等。  
　　经过一段时间的学习，你可以游刃有余的模拟手工测试来操作页面上的各种元素了。接着你需要做的就是把这些“用例”组织起来，统一来跑。  
　　那么你需要做的就是学习并使用单元测试框架，单元测试框架本身就解决了用例的组织与运行。  
　　当你写了一些“测试用例” 之后，你会发现用例中有大量重复的操作，能不能写到一个单独的文件中，需要的时候调用这些操作？当然可以，运用你的编程能力来实现这一点将非常简单。然后，你又发现每个用例中都有一些数据，这些数据也是一样的，但如果变化了修改起来非常麻烦，你也可以把他写到一个单独的文件中进行读取。  
　　接着你又遇到了新的疑问，我写的脚本（用例）都是流水式的，我怎么知道用例运行失败还是成功。那么就需要在脚本中加一些验证与断言。  
　　接着你又有了更多的想法，单元测试框架的log太简陋了，能不能生成一张漂亮的测试报告出来。我能不能定时的来跑这个脚本。能不能把每一次跑脚本的测试结果直接发到我的邮箱。能不能......  
　　为解决这些问题，你不得不学习更多的编程技术，然后你的“测试结构”会功能越来越强大，越来越灵活。产生了一定的通用性和移植性。一个有模有样的自动化测试框架诞生了。