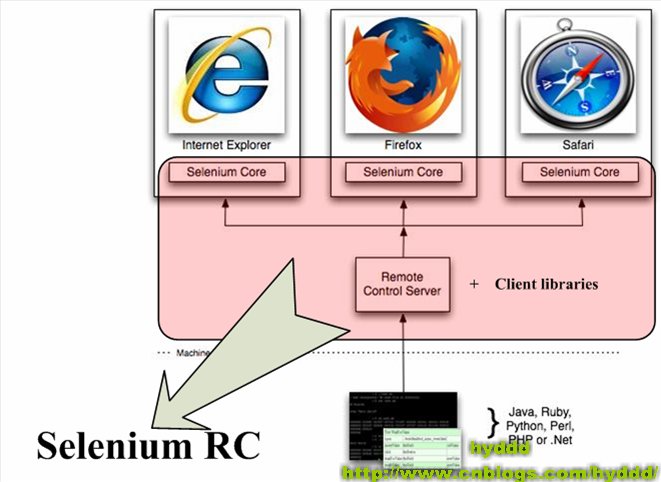
前一篇已经比较详细讲述了如何使用Selenium RC进行Web测试，但到底Selenium RC是什么?或者它由哪几部分组成呢？？

**一.Selenium RC的组成：**

关于这个问题，我拿了官网上的一幅图来说明这个问题。



Selenium RC主要由两部分组成：

**（1）.Selenium Server:**



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**，**Selenium 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。

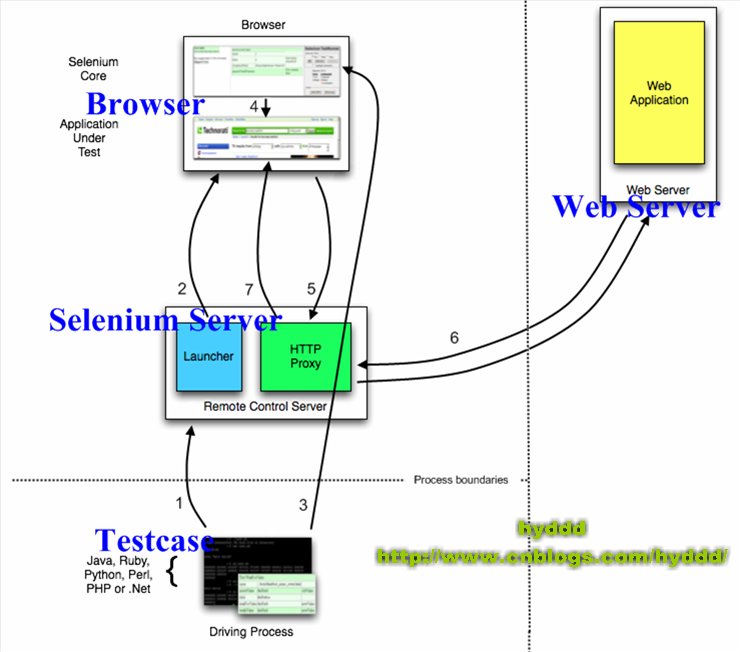
**（2）.Client Libraries:**

写测试案例时用来控制Selenium Server的库。



**二.Selenium RC与Testcase的关系**

先看下图：



(1).测试案例（Testcase）通过Client Lib的接口向Selenium Server发送Http请求，要求和Selenium Server建立连接。

为什么要通过发送Http请求控制Selenium Server而不采用其他方式呢？从上文可以看出，Selenium Server是一个独立的中间服务器（确切地说是代理服务器），它可以架设在其他机器上！所以测试案例通过发送HTTP请求去控制Selenium Server是很正常的。

(2).Selenium Server的Launcher启动浏览器，把Selenium Core加载入浏览器页面当中，并把浏览器的代理设置为Selenium Server的Http Proxy。

(3).测试案例通过Client Lib的接口向Selenium Server发送Http请求，Selenium Server对请求进行解析，然后通过Http Proxy发送JS命令通知Selenium Core执行操作浏览器的动作。

(4).Selenium Core接收到指令后，执行操作。

(5).浏览器收到新的页面请求信息（因为在(4)中，Selenium Core的操作可能引发新的页面请求），于是发送Http请求，请求新的Web页面。  
由于Selenium Server在启动浏览器时做了手脚，所以Selenium Server会接收到所有由它启动的浏览器发送的请求。

(6).Selenium Server接收到浏览器的发送的Http请求后，自己重组Http请求，获取对应的Web页面。

(7).Selenium Server的Http Proxy把接收的Web页面返回给浏览器。

为什么Selenium RC中的Selenium Server需要以这种代理服务器的形式存在呢？下一篇继续介绍:>