

Selenium RC, Selenium GRID y JUnit

Fecha de creación: 2 de diciembre de 2011.

Versión: 1.0.

Autor: Alejandro L. Zuvic, QA Tester, QAustral.

Introducción

El presente documento es una guía de instalación y utilización general de las herramientas Selenium RC, Selenium Grid y JUnit. El mismo no pretende ser un manual extensivo sobre las herramientas, sino una introducción al uso de las mismas.

Nota: en siguientes versiones del documento se explicará en mayor detalle como ejecutar test cases propios con Selenium Grid (en el actual documento se explica con un demo incluido en la herramienta), como ejecutar test cases paralelos en una red de PCs mediante Selenium Grid y se dará mayor detalle en las características de Junit.

Selenium RC

Qué es Selenium RC

Selenium Remote Control es una herramienta de testing que permite ejecutar tests automáticos para aplicaciones web escritos en virtualmente cualquier lenguaje de programación. Selenium RC ejecuta los tests contra cualquier sitio web bajo el protocolo HTTP, utilizando las capacidades JavaScript del navegador web.

Composición y funcionamiento de Selenium RC

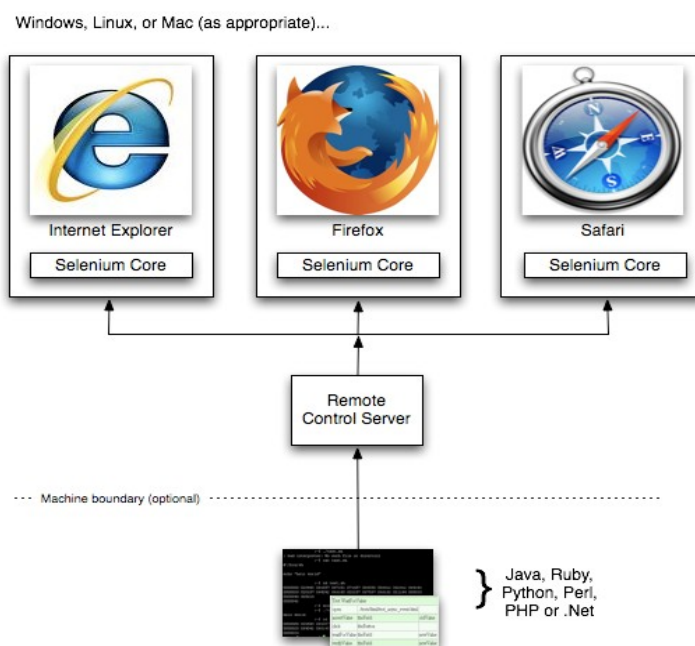
Selenium Remote Control se compone de dos partes principales:

1. Un servidor con la capacidad de abrir y cerrar navegadores, y que actúa como un proxy HTTP para servir las peticiones desde los tests.
2. Librerías para poder escribir las pruebas en diferentes lenguajes.

El test es programado en el lenguaje de la preferencia del tester. Durante la ejecución de los tests se envían los comandos en Selenese al servidor de Remote Control, que, como su nombre indica cumple función de control remoto para Selenium Core. Selenium RC envía las peticiones HTTP a Selenium Core, que finalmente ejecuta cada comando mediante instrucciones JavaScript directamente en el navegador.

Como se aprecia, la instancia final de la ejecución de los tests es peticionar las instrucciones necesarias en JavaScript. Por lo tanto, la ejecución de los tests puede ser practicada sobre cualquier navegador con capacidades JavaScript (en la actualidad, virtualmente cualquiera).

Una representación visual permitirá entender con mayor claridad el concepto.



Instalación

Como se mencionó anteriormente, Selenium RC requiere la utilización de dos componentes que se deben descargar de la página de SeleniumHQ (<http://seleniumhq.org/download/>).

- Selenium Server (Selenium RC).
- Drivers para los clientes de Selenium.

Instalación de Selenium Server/Selenium RC:

1. Descargar la última versión desde la página de descarga de SeleniumHQ. Al momento de escribir este manual, la última versión liberada es la 2.14.0.
2. Mover el archivo descargado (comúnmente “selenium-server-standalone-x.x.x.jar”) a la ubicación deseada.

El archivo descargado contiene tanto el servidor como Selenium RC y Selenium Core.

Instalación de los drivers para clientes Selenium:

Los drivers para clientes Selenium no son más que las librerías necesarias para brindarle capacidad de comunicación entre el lenguaje de programación deseado y Selenium RC. Dichas librerías agregan las clases y métodos necesarios para programar, compilar y ejecutar las pruebas.

Aunque existen drivers para multitud de lenguajes (C#, Ruby on Rails, Python, Java, entre otros) los ejemplos de instalación y creación de tests para este manual se realizarán en Java utilizando Eclipse. El proceso para otros lenguajes y entornos de desarrollo es similar.

Para instalar los drivers Java es necesario contar con el Java Development Kit 1.5 o superior (para Windows, se pueden descargar desde www.oracle.com, Mac OS Leopard y Snow Leopard poseen un JDK personalizado compatible, para Linux la instalación varía entre distribuciones).

Para conocer la versión de JDK instalada abrir una línea de comandos y ejecutar “java -version”.

Para instalar los drivers de Java:

1. Descargar los drivers deseados (en este caso, Java) desde la página de SeleniumHQ (<http://seleniumhq.org/download/>), sección “Selenium Client Drivers”.
2. Descomprimir el archivo descargado (comúnmente “selenium-java-x.x.x.zip”) en la ubicación deseada.
3. Importar las librerías descomprimidas en el proyecto Java que contiene los tests. Más adelante se explicará cómo crear los proyectos para escribir los tests, por el momento, es importante entender la lógica del proceso. En siguientes secciones se requerirá seguir estos pasos.
 1. Expandir el proyecto abierto en el explorador de paquetes de Eclipse.
 2. Hacer click con el botón derecho sobre las librerías referenciadas (“Referenced Libraries”).
 3. Seleccionar la opción Construir ruta (“Build path”) \ Configurar ruta (“Configure buildpath”).
 4. En el cuadro de diálogo abierto seleccionar “Agregar JARs externos” (“Add external JARs”).
 5. Seleccionar el archivo “selenium-java-x.x.x.jar” descomprimido anteriormente.
 6. Las librerías deberían estar agregadas.

Junit

Qué es Junit

JUnit es un framework de testing unitario para Java. Aunque se pueden ejecutar las pruebas automatizadas utilizando Selenium RC y las librerías del driver necesario, JUnit provee mayores ventajas al agregar logs, registros y mayor claridad en el código generado.

Por ser un framework específicamente desarrollado para Java, la utilización en otros lenguajes es imposible. Sin embargo, existen alternativas similares para otros lenguajes, por ejemplo CppUnit para C++ o NUnit para .NET.

Instalación

JUnit se distribuye empaquetado en un archivo .jar y se agrega a la IDE como librerías referenciadas. En Eclipse, el proceso de instalación es el siguiente.

- Descargar JUnit desde la página oficial (<http://www.junit.org/>) o directamente desde los repositorios de GitHub (<https://github.com/KentBeck/junit/downloads>). Se recomienda descargar la opción básica o el .jar sin hamcrest.
- Una vez descargado el archivo .jar (comúnmente “junitx.x.jar”) se debe importar como librerías a nuestra IDE, en este caso Eclipse:
 1. Expandir el proyecto abierto en el explorador de paquetes de Eclipse.
 2. Hacer click con el botón derecho sobre las librerías referenciadas (“Referenced Libraries”).
 3. Seleccionar la opción Construir ruta (“Build path”) \ Configurar ruta.
 4. En el cuadro de diálogo abierto seleccionar “Agregar JARs externos” (“Add external JARs”).
 5. Seleccionar el archivo “junitx.x.jar” descargado anteriormente.
 6. Las librerías deberían estar agregadas.

Con Selenium RC instalado y las librerías de los drivers de Selenium y JUnit agregadas, ya es posible ejecutar tests contra Remote Control.

Creación de tests en Java/JUnit

Existen dos maneras de obtener un test en Java compatible con Selenium RC:

- Crear el script necesario en Selenium IDE y exportarlo como test JUnit (explicado en el “Manual de Selenium IDE”).
- Programar el test desde cero. Para explicar esta última opción utilizaremos un código ejemplo básico. Se requieren conocer los conceptos elementales de los comandos de Selenese (explicados en el “Manual de Selenium IDE”) y conocimientos básicos de Java. Se desglosará el código explicando cada parte del mismo. Para esto se creará un nuevo proyecto en Java, proyecto al cual deben agregarse las librerías de los drivers para Selenium y JUnit (explicado anteriormente). Se entiende que el lector tiene conocimientos básicos en la utilización de Eclipse (u otra IDE similar) y de Java.

Código ejemplo

```
package com.example.tests;

import com.thoughtworks.selenium.*;
import org.junit.After;
import org.junit.Before;
import org.junit.Test;
import java.util.regex.Pattern;

@SuppressWarnings("deprecation")
public class TestWiki extends SeleneseTestCase{
    @Before

    public void setUp() throws Exception {
        selenium = new DefaultSelenium("localhost", 4444, "*firefox",
"http://en.wikipedia.org/");
        selenium.start();
    }

    @Test
    public void testWiki() throws Exception {
        selenium.open("/wiki/Main_Page");
        selenium.waitForPageToLoad("50000");
        selenium.type("id=searchInput", "Technological singularity");
        selenium.click("id=searchButton");
        selenium.waitForPageToLoad("50000");
    }

    @After
    public void tearDown() throws Exception {
        selenium.stop();
    }
}
```

Explicación

```
package com.example.tests;
```

Paquete donde se encuentra nuestro test.

```
import com.thoughtworks.selenium.*;  
import org.junit.After;  
import org.junit.Before;  
import org.junit.Test;  
import java.util.regex.Pattern;
```

Se importan los paquetes de Selenium driver y JUnit.

```
@SuppressWarnings("deprecation")
```

Se suprimen los warnings por métodos deprecados (es útil hacerlo si existe una versión más reciente de JUnit o Selenium RC drivers).

```
public class TestWiki extends SeleneseTestCase{
```

Se declara la clase que contiene el test. Se extienden los métodos de *SeleneTestCase* necesarios.

```
@Before
```

```
public void setUp() throws Exception {  
    selenium = new DefaultSelenium("localhost", 4444, "*firefox",  
    "http://en.wikipedia.org/");  
    selenium.start();  
}
```

La etiqueta *@Before* le indica a JUnit que la subsiguiente porción de código debe ejecutarse antes de ejecutar el test. Entiéndase como “la preparación antes del test”.

El método *setUp()* inicializa el navegador y lo ejecuta.

La primera línea inicializa el navegador en forma de una instancia de *DefaultSelenium*. *DefaultSelenium* requiere los argumentos *server* (servidor de Selenium RC, en este caso local), el *puerto de escucha* (el puerto que utiliza Selenium RC para recibir comandos, en este caso 4444), el *navegador* (en este caso Mozilla Firefox, customizable mediante “*custom <Ruta de acceso al ejecutable del browser>”), *URL base* (URL sobre la que se ejecuta el test, en este caso la página inglesa de Wikipedia).

El método *start()* inicia el navegador.

```
@Test
```

```
public void testWiki() throws Exception {  
    selenium.open("/wiki/Main_Page");  
    selenium.waitForPageToLoad("50000");  
    selenium.type("id=searchInput", "Technological singularity");  
    selenium.click("id=searchButton");  
    selenium.waitForPageToLoad("50000");  
}
```

La etiqueta *@Test* le indica a JUnit que las subsiguientes líneas de código son parte del test y que deben ser ejecutadas como tal.

El método *testWiki()* contiene las instrucciones necesarias para la ejecución del test. Como se ve, cada comando en Selenium IDE posee una correspondencia con un método de la clase *DefaultSelenium*, en este caso instanciado como *selenium*. No se explicarán en mayor detalle estas líneas ya que los comandos ejecutados fueron explicados en el Manual de Selenium IDE.

```
@After
```

```
public void tearDown() throws Exception {  
    selenium.stop();  
}
```

La etiqueta *@After* le indica a JUnit que las subsiguientes líneas de código dan por finalizado el test, ejecutando el método *tearDown()*, necesario para la detención del test.

El método *stop()* de la clase *DefaultSelenium* cierra el navegador y da por finalizado el test.

Analizando el código, se entiende que al ejecutar el test case: se abrirá una instancia de Mozilla Firefox con la página inicial de Wikipedia en inglés, se dirigirá el navegador a la portada de dicha página, se escribirá “Technological singularity” en el campo de búsqueda y se hará click en el botón de búsqueda, cargando así el artículo correspondiente.

Finalmente, se da por terminado el test, deteniendo la ejecución del navegador y devolviendo JUnit las estadísticas de ejecución.

Ejecución del test

Para ejecutar el test, se debe:

- Ejecutar el servidor de Selenium RC, que como se explicó, recibe las instrucciones en Selenese del test JUnit y las envía como peticiones a Selenium Core, que ejecuta dichas peticiones como código JavaScript.
- Ejecutar el test JUnit programado anteriormente.

Ejecutar Selenium RC:

1. En Windows, ejecutar la línea de comandos (Ejecutar, “cmd.exe”).
2. Navegar hasta la ubicación donde se encuentra el archivo correspondiente al server de Selenium RC (comúnmente “selenium-standalone-x.x.x.jar”).
3. Ejecutar el comando “java -jar selenium-standalone-x.x.x.jar”. Se puede agregar el argumento *-debug* para iniciar el servidor en modo debugger.
4. Selenium RC y su servidor deberían estar ejecutándose.

Nota: en otros sistemas operativos, el proceso es similar, simplemente se debe utilizar la terminal o consola provista con el sistema operativo en lugar de la línea de comandos de Windows.

Es útil crear un batch script básico para evitar este proceso cada vez que se requiera iniciar el server.

Ejecutar el test JUnit:

1. Seleccionar la clase donde se encuentra el test (en nuestro ejemplo anterior “TestWiki”).
2. Compilar y ejecutar. Alternativamente:
 - a. Se puede compilar y ejecutar mediante línea de comando con la instrucción “java TestWiki.jar”.

Como se ve, y contrariamente a la mayoría de los programas desarrollados en Java, en ningún paso del proceso se requiere la ejecución de un método *main()*. JUnit simplemente no requiere la utilización del método principal ya que reconoce el orden de ejecución mediante las etiquetas *@Before*, *@Test* y *@After*.

Si se desea ejecutar el test mediante un método *main()* es posible instanciar la clase del test y ejecutar los métodos *setUp()*, “*test*”, *tearDown()* mediante la llamada directa a dichos métodos. Esto, sin embargo, no ejecuta los procesos de diagnóstico de JUnit. La escritura de un método *main()* que ejecute JUnit excede el nivel de este manual.

Selenium Grid

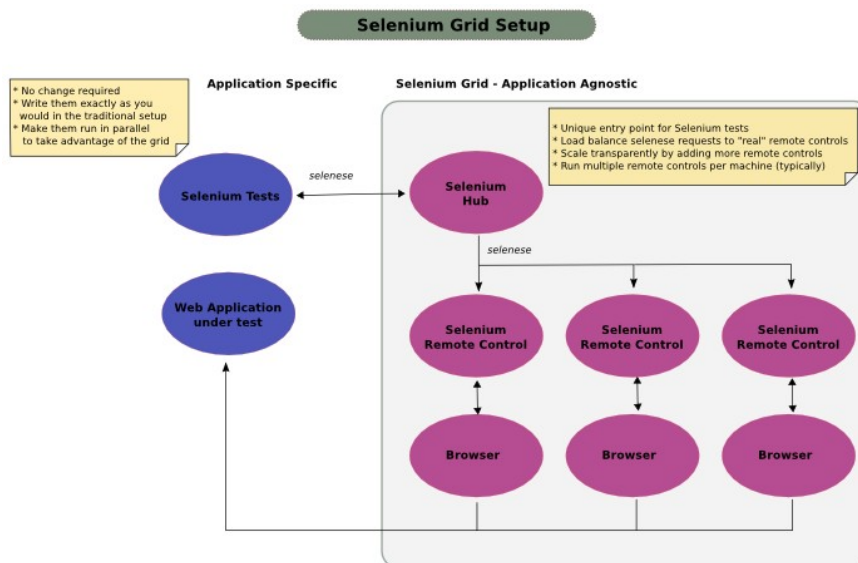
Qué es Selenium Grid

Selenium Grid es una herramienta que acelera el proceso de testing funcional para aplicaciones web aprovechando la infraestructura existente.

Selenium Grid permite ejecutar múltiples instancias de Selenium RC en paralelo, acelerando el proceso de testing y aprovechando mejor el hardware existente.

Cómo funciona Selenium Grid

Selenium Grid funciona ejecutando un componente conocido como Selenium Hub, que contiene cada instancia y direcciona las peticiones de cada test a la instancia apropiada de Selenium RC. Un diagrama explicará esto en mayor detalle.



La transparencia de Selenium Grid permite la ejecución de los casos de prueba programados bajo JUnit sin cambios en los mismos.

Instalación de Selenium Grid

Para ejecutar Selenium Grid se requiere:

- JDK 1.5 o mayor. Ya se explicó como verificar la versión de Java instalada y su correspondiente instalación.
- Ant 1.7 o superior. A continuación se explicará como instalar Ant.

Instalación de Ant:

Para instalar Ant 1.7 sobre Windows existen dos posibilidades:

- Descargar la distribución binaria.
- Utilizar el instalador Winant.

Instalar mediante distribución binaria:

1. Descargar la distribución binaria desde la página de Selenium Grid (http://selenium-grid.seleniumhq.org/get_started.html).
2. Descomprimir el archivo en la ubicación deseada.
3. Abrir una línea de comando de Windows.
4. Ejecutar el comando “set HOME_ANT=<ruta a Ant>”.
5. Ejecutar el comando “set HOME_JAVA=<ruta del JDK>”.
6. Chequear la instalación mediante el comando “ant -version”.

Instalar mediante Winant:

1. Descargar Winant desde la página oficial (<http://code.google.com/p/winant/>).
2. Ejecutar el instalador descargado.
3. Reiniciar la PC.
4. Chequear la instalación de Ant abriendo una línea de comando y ejecutando “ant -version”.

Instalación de Selenium Grid:

1. Descargar la distribución binaria desde la página oficial (<http://selenium-grid.seleniumhq.org/download.html>).
2. Descomprimir el contenedor en la ubicación deseada.
3. Abrir una línea de comando de Windows.
4. Navegar hasta el directorio donde se encuentra descomprimido Selenium Grid.
5. Validar la instalación mediante el comando “ant sanity -check”.

En Mac OS X Leopard o superior no se requiere instalación de Ant, ya que viene incluido en el sistema operativo. En Linux los pasos de instalación dependen de la distribución.

La instalación de Selenium Grid en otros sistemas operativos es similar. En lugar de utilizar la línea de comandos, se requiere utilizar la terminal/consola propia del sistema.

Ejecución de tests mediante Selenium Grid

Para explicar la ejecución de un test mediante Selenium Grid utilizaremos el demo incluido en Selenium Grid.

Ejecución en secuencia:

1. Abrir una nueva Terminal/Línea de comandos de Windows.
2. Navegar hasta el directorio de Selenium Grid.
3. Ejecutar el comando “ant launch-hub”. Esto ejecutará el hub que se mencionó en el diagrama.
4. Chequear que el hub está corriendo. Mediante un navegador web ir a la dirección <http://localhost:4444/console>.
5. Abrir una nueva Terminal/Línea de comandos de Windows.
6. Ejecutar “ant launch-remote-control”. Esto ejecutará una instancia de Selenium RC.
7. En el navegador ahora debe aparecer un Remote Control disponible mediante una tabla de la forma:

Host	Port	Environment
localhost	5555	*firefox

8. Abrir una nueva Terminal/Línea de comando de Windows.
9. Ejecutar “ant run-demo-in-sequence”. Esto ejecutará el demo en secuencia.

Cuatro tests Selenium se ejecutarán en secuencia contra el Remote Control. Esto es parte del demo incluido en Selenium Grid. Para saber que hace el test demo, verificar el código en el directorio demo/sub-project.

Ejecución en paralelo:

Para ejecutar el demo en paralelo, seguir las instrucciones de la sub-sección anterior hasta el paso siete.

Luego:

1. Abrir tres Terminales/Líneas de comando de Windows.
2. Ejecutar, en cada una de ellas los comandos:
“ant -Dport=5556 launch-remote-control
ant -Dport=5557 launch-remote-control
ant -Dport=5558 launch-remote-control”
Esto ejecutará tres Remote Controls con los puertos de escucha en 5556, 5557, 5558. Eso sumará un total de cuatro Remote Controls (si se siguieron los pasos anteriores).
Los cuatro RC deberían verse en el navegador en la dirección <http://localhost:4444/console> en la forma:

Host	Port	Environmen t
localhost	5555	*firefox
localhost	5556	*firefox
localhost	5557	*firefox
localhost	5558	*firefox

3. Abrir una nueva Terminal/Línea de comandos de Windows.

4. Ejecutar “ant run-demo-in-parallel”.

Al estar ejecutándose cuatro tests sobre cuatro instancias de Selenium RC, el tiempo de prueba debería ser significativamente menor.

Nota: en siguientes versiones del documento se explicará como correr tests personalizados y como correr tests en paralelo utilizando varias PCs en red.

Tel. Bs As, Argentina: 011 53685847
Tel. San Francisco EEUU: 001 14157041515
Tel. Londres UK: 00442033183814



Av. Rafael Nuñez 3612
2do piso – Oficina 11.
Córdoba –Argentina.
TEL: 0054 - 0351 - 4817951