



EBook Gratis

APRENDIZAJE selenium-grid

Free unaffiliated eBook created from
Stack Overflow contributors.

#selenium-
grid

Tabla de contenido

Acerca de.....	1
Capítulo 1: Empezando con la red de selenio.....	2
Observaciones.....	2
Examples.....	2
¿Qué es la rejilla de selenio?.....	2
¿Qué es un hub y nodo?.....	2
Capítulo 2: Configuración de cuadrícula de selenio.....	4
Observaciones.....	4
Descargas.....	4
h11.....	4
Conductores.....	4
Navegadores.....	4
Examples.....	4
Instalación o configuración.....	4
Configurar el Hub.....	4
Configurar los nodos.....	6
Cosas a tener en cuenta.....	7
Configuracion json.....	8
Configuración y uso en C #.....	9
Configuración.....	9
Microsoft Edge.....	9
Cromo.....	9
Firefox.....	10
Ópera.....	10
Configuración de Json y C # múltiples navegadores.....	11
Configuración.....	11
Microsoft Edge.....	11
Cromo.....	12
Firefox.....	12
Ópera.....	12

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [selenium-grid](#)

It is an unofficial and free selenium-grid ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official selenium-grid.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capítulo 1: Empezando con la red de selenio

Observaciones

Esta sección proporciona una descripción general de qué es la red de selenio y por qué un desarrollador puede querer usarlo.

También debe mencionar los temas importantes dentro de la cuadrícula de selenio y vincular a los temas relacionados. Dado que la Documentación para selenio-cuadrícula es nueva, es posible que deba crear versiones iniciales de esos temas relacionados.

Examples

¿Qué es la rejilla de selenio?

Selenium-Grid es una configuración de *Hub & Node* que le permite ejecutar sus pruebas en diferentes máquinas con diferentes combinaciones de navegador en paralelo. Es decir, ejecutar varias pruebas al mismo tiempo en *diferentes máquinas que ejecutan diferentes navegadores en diferentes sistemas operativos* . En otras palabras, Selenium Grid admite la ejecución de pruebas en **Entorno distribuido** .

Cuando usarlo

- **Para ejecutar sus pruebas en múltiples navegadores, múltiples versiones de navegadores y navegadores que se ejecutan en diferentes sistemas operativos.** Esto asegurará que la aplicación que está probando sea totalmente compatible con una amplia gama de combinaciones de navegador y sistema operativo.
- **Para reducir el tiempo que tarda la suite de pruebas en completar un pase de prueba.** Digamos que configura su Grid para ejecutar 8 pruebas a la vez, su ejecución terminará 8 veces más rápido en comparación con su ejecución normal.

Selenium-Grid se utiliza para acelerar la ejecución de una prueba utilizando varias máquinas para ejecutar pruebas en paralelo.

¿Qué es un hub y nodo?

El hub

- El *concentrador* es el motor principal / punto central de toda la configuración, punto donde se conectan todos los nodos.
- *El hub* debe funcionar solo en una sola máquina.
- Solo debe haber 1 concentrador en funcionamiento donde se cargan todas las pruebas.
- Las pruebas se ejecutarán en las máquinas donde se ejecuta hub, pero puede ver los navegadores en las máquinas de nodo.

Los nodos

- Los nodos son las instancias (máquinas) que ejecutarán las pruebas que se cargan en el concentrador.
- No hay limitaciones en las máquinas de nodos, un usuario puede configurar un número de nodos.
- Los nodos se pueden iniciar en diferentes máquinas con diferentes combinaciones de sistema operativo y navegador.
- Las máquinas que ejecutan los nodos pueden tener configuraciones diferentes o iguales a las de Hub Machine.

Lea **Empezando con la red de selenio en línea**: <https://riptutorial.com/es/selenium-grid/topic/8701/empezando-con-la-red-de-selenio>

Capítulo 2: Configuración de cuadrícula de selenio

Observaciones

Descargas

Este capítulo contiene descargas útiles como los controladores web y el enlace a los navegadores.

Conductores

Coloca todos los drivers en tu variable de camino.

- [Driver cromo](#)
- [Controlador de FireFox](#)
- [Controlador Microsoft Edge](#)
- [Conductor de ópera](#)

Navegadores

- [Cromo](#)
- [FireFox](#)
- [Microsoft Edge](#)
- [Ópera](#)

Examples

Instalación o configuración

Antes de configurar una cuadrícula de Selenium, debe asegurarse de tener *Java* instalado y configurado en la ruta del entorno de su computadora.

Configurar el Hub

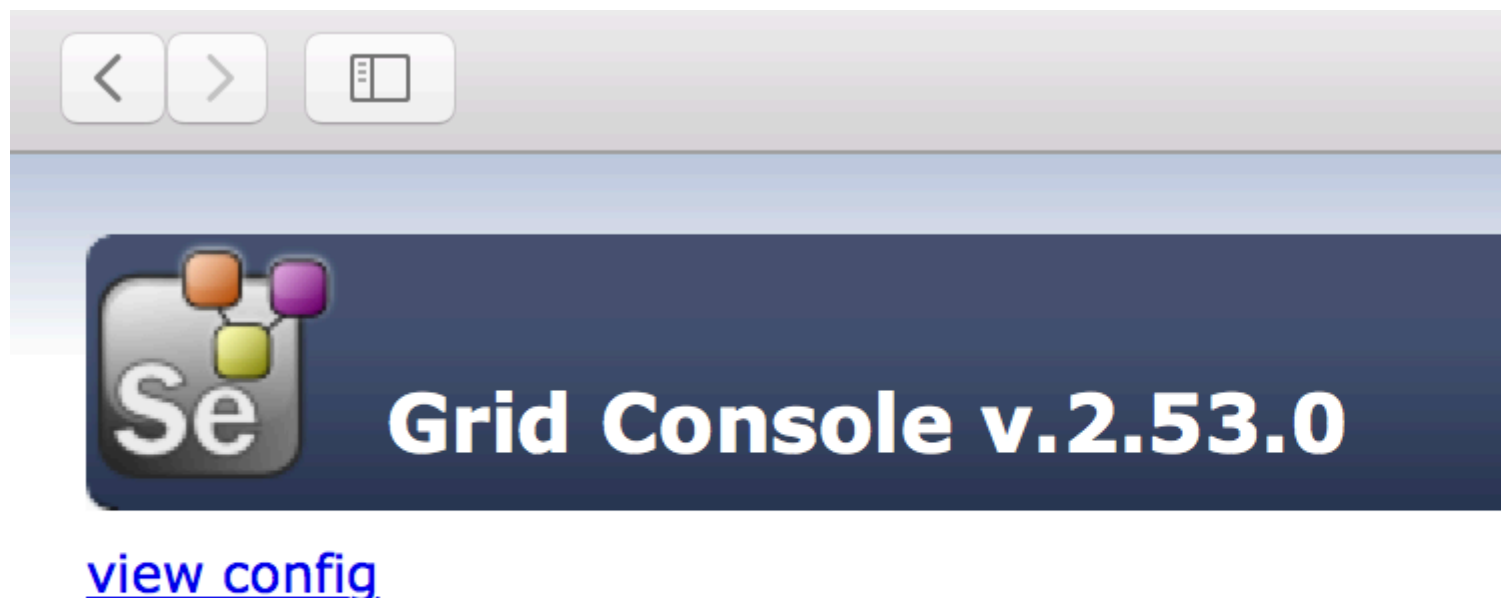
- Descargue la última versión estable de [Selenium Server](#) .
- Inicie el símbolo del sistema y navegue hasta la ubicación en la que colocó el archivo jar del servidor Selenium.
- Escriba: (FYI: su número de versión puede ser diferente al mío) `java -jar selenium-server-standalone-2.53.0.jar -role hub`

- Ahora debería verse algo como esto:

```
→ jars java -jar selenium-server-standalone-2.53.0.jar
18:47:47.018 INFO - Launching Selenium Grid hub
2017-01-03 18:47:47.524:INFO::main: Logging initialized
18:47:47.533 INFO - Will listen on 4444
18:47:47.566 INFO - Will listen on 4444
2017-01-03 18:47:47.569:INFO:osjs.Server:main: jetty-
2017-01-03 18:47:47.590:INFO:osjs.ContextHandler:mai
2017-01-03 18:47:47.612:INFO:osjs.ServerConnector:mai
2017-01-03 18:47:47.612:INFO:osjs.Server:main: Starte
18:47:47.613 INFO - Nodes should register to http://1
18:47:47.613 INFO - Selenium Grid hub is up and runni
```

Básicamente, lo que sucedió fue que el servidor web de selenio comenzó y ahora está escuchando en un puerto, en este caso por defecto **4444** (para su información: este número de puerto se puede cambiar pasando el parámetro *-port* seguido del número de puerto en el que desea ejecutar el servidor).

- Ahora abra un navegador y navegue a <http://localhost:4444/grid/console>
- Si todo funciona, el servidor debería aparecer y verías algo como esto:



A continuación, necesitamos configurar algunas máquinas de nodo.

Configurar los nodos

- Al igual que descargamos [Selenium Server](#) for **Hub** , también necesitamos descargarlo en todas nuestras máquinas **Node** .
- Una vez que tenga el archivo jar de Selenium-server en la máquina del nodo, navegue hasta el directorio donde se descarga el archivo jar y abra el indicador de cmd.
- Escriba: `java -jar selenium-server-standalone-2.53.0.jar -nodo de rol -hub http://hubIP:4444 / grid / register`
 - **hubIP** : - en caso de que el hub y los nodos se estén ejecutando en una máquina diferente
 - **localhost** : - en caso de que el hub y los nodos se ejecuten en la misma máquina

Como puede ver, el nodo ahora está registrado en el concentrador, el nodo predeterminado comienza en **-port 5555** pero puede cambiarlo utilizando el parámetro **-port** seguido del número de puerto.

```
→ jars java -jar selenium-server-standalone-2.53.0.jar
21:28:54.200 INFO - Launching a Selenium Grid node
21:28:54.525 INFO - Java: Oracle Corporation 25.31-b0
21:28:54.525 INFO - OS: Mac OS X 10.12.2 x86_64
21:28:54.529 INFO - v2.53.0, with Core v2.53.0. Built
21:28:54.569 INFO - Driver provider org.openqa.selenium
registration capabilities Capabilities [{ensureCleanS
21:28:54.569 INFO - Driver provider org.openqa.selenium
registration capabilities Capabilities [{browserName=
21:28:54.569 INFO - Driver class not found: com.opera
21:28:54.569 INFO - Driver provider com.opera.core.sy
21:28:54.571 INFO - Driver class not found: org.openc
21:28:54.571 INFO - Driver provider org.openqa.selenium
21:28:54.604 INFO - Selenium Grid node is up and read
21:28:54.628 INFO - Starting auto registration thread
21:28:54.628 INFO - Registering the node to the hub:
21:28:54.650 INFO - The node is registered to the hub
```

Si todo funciona como se esperaba, ahora debería ver la dirección IP del nodo que acaba de iniciar y registrar en la vista de la consola central:



Grid Console v.2.53.0

DefaultRemoteProxy (version : 2.53.0)

id : http://[redacted].1[redacted].1[redacted]:5555, OS : MAC

Browsers

Configuration

Remote Control (legacy)

v:

v:

v:

WebDriver

v:

v:

v:

[view config](#)

Cosas a tener en cuenta

- Si no especificamos el `seleniumProtocol` , Node se registrará tanto con Remote Control (Legacy) como con Webdriver Protocol (como se ve en la captura de pantalla anterior).
- Si el tipo de navegador y el número de instancias no se mencionan, Node lanzará 5

instancias de Firefox, 5 Instancia de Chrome y 1 Instancia de controlador de IE.

Eso es todo lo que necesitarías hacer para una Selenium Grid en funcionamiento.

Configuración json

Una configuración de ejemplo para un hub:

```
java -jar selenium-server-standalone-<version>.jar -role hub -hubConfig hubConfig.json
```

```
{
  "_comment" : "Configuration for Hub - hubConfig.json",
  "host": ip,
  "maxSessions": 5,
  "port": 4444,
  "cleanupCycle": 5000,
  "timeout": 300000,
  "newSessionWaitTimeout": -1,
  "servlets": [],
  "prioritizer": null,
  "capabilityMatcher": "org.openqa.grid.internal.utils.DefaultCapabilityMatcher",
  "throwOnCapabilityNotPresent": true,
  "nodePolling": 180000,
  "platform": "WINDOWS"
}
```

Una configuración de ejemplo para un nodo

```
java -jar selenium-server-standalone-<version>.jar -role node -nodeConfig nodeConfig.json
```

```
{
  "capabilities":
  [
    {
      "browserName": "opera",
      "platform": "WINDOWS",
      "maxInstances": 5,
      "seleniumProtocol": "WebDriver",
      "webdriver.opera.driver": "C:/Selenium/drivers/operadriver.exe",
      "binary": "C:/Program Files/Opera/44.0.2510.1159/opera.exe"
    },
    {
      "browserName": "chrome",
      "platform": "WINDOWS",
      "maxInstances": 5,
      "seleniumProtocol": "WebDriver",
      "webdriver.chrome.driver": "C:/Selenium/drivers/chromedriver.exe",
      "binary": "C:/Program Files/Google/Chrome/Application/chrome.exe"
    },
    {
      "browserName": "firefox",
      "platform": "WINDOWS",
      "maxInstances": 5,
      "seleniumProtocol": "WebDriver",
      "webdriver.gecko.driver": "C:/Selenium/drivers/geckodriver.exe",
      "binary": "C:/Program Files/Mozilla Firefox/firefox.exe"
    }
  ],
}
```

```
"proxy": "org.openqa.grid.selenium.proxy.DefaultRemoteProxy",
"maxSession": 5,
"port": 5555,
"register": true,
"registerCycle": 5000,
"hub": "http://localhost:4444",
"nodeStatusCheckTimeout": 5000,
"nodePolling": 5000,
"role": "node",
"unregisterIfStillDownAfter": 60000,
"downPollingLimit": 2,
"debug": false,
"servlets" : [],
"withoutServlets": [],
"custom": {}
}
```

Configuración y uso en C

Configuración

En los siguientes párrafos habrá un ejemplo por navegador para la configuración en Json y la configuración en C #.

Este ejemplo espera que tenga todos los controladores en su variable de ruta y los navegadores instalados.

Microsoft Edge

Código C # para crear un controlador web remoto

```
// Defining webdriver variable
RemoteWebDriver _webDriver;
// Creating Capabilities and choosing browser
capabilities = DesiredCapabilities.Edge();
// Setting platform
capabilities.Platform = new Platform(PlatformType.Windows);
// Requesting remote webdriver
_webDriver = new RemoteWebDriver(_gridServerUri, capabilities);
```

Configuración del nodo en Json

```
{
  "browserName": "MicrosoftEdge",
  "platform": "WINDOWS",
  "maxInstances": 1,
  "seleniumProtocol": "WebDriver"
}
```

Cromo

Código C # para crear un controlador web remoto

```
// Defining webdriver variable
RemoteWebDriver _webdriver;
// Creating Capabilities and choosing browser
capabilities = DesiredCapabilities.Chrome();
// Setting platform
capabilities.Platform = new Platform(PlatformType.Windows);
// Requesting remote webdriver
_webdriver = new RemoteWebDriver(_gridServerUri, capabilities);
```

Configuración del nodo en Json

```
{
  "browserName": "chrome",
  "platform": "WINDOWS",
  "maxInstances": 5,
  "seleniumProtocol": "WebDriver"
}
```

Firefox

Código C # para crear un controlador web remoto

```
// Defining webdriver variable
RemoteWebDriver _webdriver;
// Creating Capabilities and choosing browser
capabilities = DesiredCapabilities.Firefox();
// Setting platform
capabilities.Platform = new Platform(PlatformType.Windows);
// Requesting remote webdriver
_webdriver = new RemoteWebDriver(_gridServerUri, capabilities);
```

Configuración del nodo en Json

```
{
  "browserName": "firefox",
  "platform": "WINDOWS",
  "maxInstances": 5,
  "seleniumProtocol": "WebDriver"
}
```

Ópera

Código C # para crear un controlador web remoto **Esto es para OperaChromium**

```
// Defining webdriver variable
RemoteWebDriver _webdriver;
// Creating Capabilities
capabilities = new DesiredCapabilities();
// Setting platform
capabilities.Platform = new Platform(PlatformType.Windows);
// Choosing browser
capabilities.SetCapability(CapabilityType.BrowserName, "operablank");
// Requesting remote webdriver
_webdriver = new RemoteWebDriver(_gridServerUri, capabilities);
```

Configuración del nodo en Json

```
{
  "browserName": "operablink",
  "platform": "WINDOWS",
  "maxInstances": 5,
  "seleniumProtocol": "WebDriver"
}
```

El tipo de plataforma puede ser uno de los siguientes:

- PlatformType.Android;
- PlatformType.Any;
- PlatformType.Linux;
- PlatformType.Mac;
- PlatformType.Unix;
- PlatformType.Vista;
- PlatformType.Windows;
- PlatformType.WinNT;
- PlatformType.XP;

Configuración de Json y C # múltiples navegadores

Configuración

En los siguientes párrafos habrá un ejemplo por navegador para la configuración en Json y la configuración en C #.

Este ejemplo espera que tenga todos los navegadores instalados y los controladores en su variable de ruta

Microsoft Edge

Código C # para crear un controlador web remoto

```
// Defining webdriver variable
RemoteWebDriver _webdriver;
// Creating Capabilities and choosing browser
capabilities = DesiredCapabilities.Edge();
// Setting platform
capabilities.Platform = new Platform(PlatformType.Windows);
// Requesting remote webdriver
_webdriver = new RemoteWebDriver(_gridServerUri, capabilities);
```

Configuración del nodo en Json

```
{
  "browserName": "MicrosoftEdge",
  "platform": "WINDOWS",
  "maxInstances": 1,
}
```

```
"seleniumProtocol": "WebDriver"
}
```

Cromo

Código C # para crear un controlador web remoto

```
// Defining webdriver variable
RemoteWebDriver _webdriver;
// Creating Capabilities and choosing browser
capabilities = DesiredCapabilities.Chrome();
// Setting platform
capabilities.Platform = new Platform(PlatformType.Windows);
// Requesting remote webdriver
_webdriver = new RemoteWebDriver(_gridServerUri, capabilities);
```

Configuración del nodo en Json

```
{
  "browserName": "chrome",
  "platform": "WINDOWS",
  "maxInstances": 5,
  "seleniumProtocol": "WebDriver"
}
```

Firefox

Código C # para crear un controlador web remoto

```
// Defining webdriver variable
RemoteWebDriver _webdriver;
// Creating Capabilities and choosing browser
capabilities = DesiredCapabilities.Firefox();
// Setting platform
capabilities.Platform = new Platform(PlatformType.Windows);
// Requesting remote webdriver
_webdriver = new RemoteWebDriver(_gridServerUri, capabilities);
```

Configuración del nodo en Json

```
{
  "browserName": "firefox",
  "platform": "WINDOWS",
  "maxInstances": 5,
  "seleniumProtocol": "WebDriver"
}
```

Ópera

Código C # para crear un controlador web remoto

```
// Defining webdriver variable
```

```
RemoteWebDriver _webDriver;  
// Creating Capabilities  
capabilities = new DesiredCapabilities();  
// Setting platform  
capabilities.Platform = new Platform(PlatformType.Windows);  
// Chosing browser  
capabilities.SetCapability(CapabilityType.BrowserName, "operablink");  
// Requesting remote webdriver  
_webDriver = new RemoteWebDriver(_gridServerUri, capabilities);
```

Configuración del nodo en Json

```
{  
  "browserName": "operablink",  
  "platform": "WINDOWS",  
  "maxInstances": 5,  
  "seleniumProtocol": "WebDriver"  
}
```

El tipo de plataforma puede ser uno de los siguientes:

```
PlatformType.Android;  
PlatformType.Any;  
PlatformType.Linux;  
PlatformType.Mac;  
PlatformType.Unix;  
PlatformType.Vista;  
PlatformType.Windows;  
PlatformType.WinNT;  
PlatformType.XP;
```

Lea Configuración de cuadrícula de selenio en línea: <https://riptutorial.com/es/selenium-grid/topic/8702/configuracion-de-cuadrícula-de-selenio>

Creditos

S. No	Capítulos	Contributors
1	Empezando con la red de selenio	Community , Paras
2	Configuración de cuadrícula de selenio	Paras , Thomas