



eBook Gratuit

APPRENEZ selenium-grid

eBook gratuit non affilié créé à partir des
contributeurs de Stack Overflow.

#selenium-
grid

Table des matières

À propos.....	1
Chapitre 1: Démarrer avec la grille de sélénium.....	2
Remarques.....	2
Exemples.....	2
Qu'est-ce que la grille de sélénium?.....	2
Qu'est-ce qu'un Hub & Node?.....	2
Chapitre 2: Configuration de la grille de sélénium.....	4
Remarques.....	4
Téléchargements.....	4
h11.....	4
Les conducteurs.....	4
Les navigateurs.....	4
Exemples.....	4
Installation ou configuration.....	4
Configurez le concentrateur.....	4
Configurer les nœuds.....	6
Choses à noter.....	7
Configuration Json.....	8
Configuration et utilisation en C #.....	9
Configuration.....	9
Microsoft Edge.....	9
Chrome.....	9
Firefox.....	10
Opéra.....	10
Configuration de plusieurs navigateurs Json et C #.....	11
Configuration.....	11
Microsoft Edge.....	11
Chrome.....	12
Firefox.....	12
Opéra.....	12

À propos

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

Chapitre 1: Démarrer avec la grille de sélénium

Remarques

Cette section fournit une vue d'ensemble de ce qu'est la grille de sélénium et pourquoi un développeur peut vouloir l'utiliser.

Il devrait également mentionner tous les grands sujets de la grille de sélénium et les relier aux sujets connexes. La documentation de la grille de sélénium étant nouvelle, vous devrez peut-être créer des versions initiales de ces rubriques connexes.

Exemples

Qu'est-ce que la grille de sélénium?

Selenium-Grid est une configuration de *Hub & Node* qui vous permet d'exécuter vos tests sur différentes machines avec différentes combinaisons de navigateurs en parallèle. C'est-à-dire exécuter plusieurs tests simultanément sur *différentes machines* exécutant *différents navigateurs* sur *différents systèmes d'exploitation*. En d'autres termes, Selenium Grid prend en charge l'exécution des tests dans l' **environnement distribué**.

Quand l'utiliser

- **Pour exécuter vos tests sur plusieurs navigateurs, plusieurs versions du navigateur et des navigateurs exécutés sur différents systèmes d'exploitation.** Cela garantira que l'application que vous testez est entièrement compatible avec une large gamme de combinaisons de navigateur et de système d'exploitation.
- **Réduire le temps nécessaire à la suite de tests pour effectuer un test.** Disons que vous configurez votre grille pour exécuter 8 tests à la fois, votre exécution se terminerait 8 fois plus vite que votre exécution normale.

Selenium-Grid est utilisé pour accélérer l'exécution d'un test en utilisant plusieurs machines pour exécuter des tests en parallèle.

Qu'est-ce qu'un Hub & Node?

Le hub

- Le *concentrateur* est le principal moteur / point central de la configuration entière, point où tous les nœuds sont connectés.
- *Hub* ne doit fonctionner que sur une seule machine.
- Il ne doit y avoir qu'un seul concentrateur sur lequel tous les tests sont chargés.
- Les tests seront exécutés sur les machines sur lesquelles le concentrateur est en cours

d'exécution, mais vous pouvez voir les navigateurs sur les machines du noeud.

Les nœuds

- Les nœuds sont les instances (machines) qui exécutent les tests chargés sur le concentrateur.
- Il n'y a pas de limitations sur les machines Node, un utilisateur peut configurer n nombre de nœuds.
- Les nœuds peuvent être lancés sur différentes machines avec différentes combinaisons de systèmes d'exploitation et de navigateurs.
- Les machines exécutant les nœuds peuvent être de configurations différentes / identiques à celles de Hub Machine.

Lire Démarrer avec la grille de sélénium en ligne: <https://riptutorial.com/fr/selenium-grid/topic/8701/demarrer-avec-la-grille-de-selenium>

Chapitre 2: Configuration de la grille de sélénium

Remarques

Téléchargements

Ce chapitre contient des téléchargements utiles comme les webdrivers et un lien vers les navigateurs

Les conducteurs

Placez tous les pilotes dans votre variable de chemin

- [Pilote chrome](#)
- [Pilote firefox](#)
- [Pilote Microsoft Edge](#)
- [Pilote d'opéra](#)

Les navigateurs

- [Chrome](#)
- [FireFox](#)
- [Microsoft Edge](#)
- [Opéra](#)

Exemples

Installation ou configuration

Avant de configurer une grille Selenium, vous devez vous assurer que *Java* est installé et configuré dans le chemin d'environnement de votre ordinateur.

Configurez le concentrateur

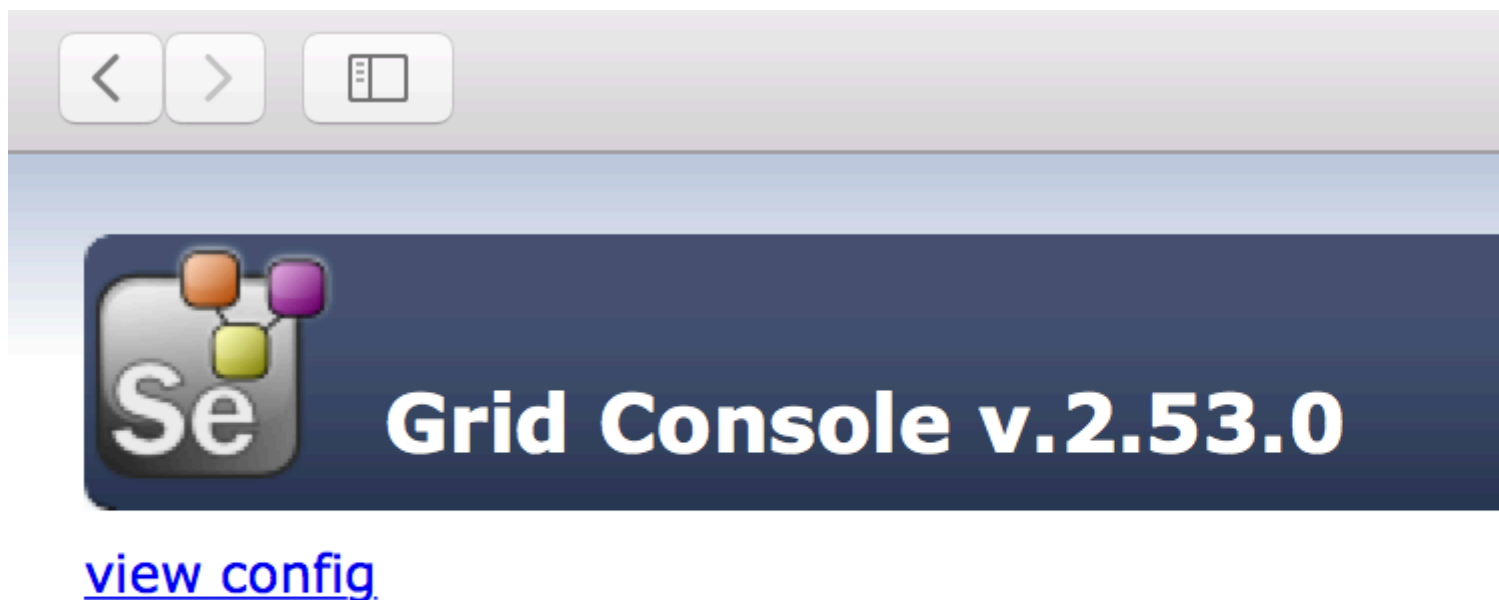
- Téléchargez la dernière version stable du [serveur Selenium](#) .
- Démarrez l'invite de commande et accédez à l'emplacement dans lequel vous avez placé le fichier jar du serveur Selenium.
- Tapez: (FYI: votre numéro de version peut être différent du mien) `java -jar selenium-server-standalone-2.53.0.jar -role hub`

- Il devrait maintenant ressembler à ceci:

```
→ 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:osjsh.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
```

Fondamentalement, le serveur Web de sélénium a démarré et écoute maintenant sur un port - dans ce cas, par défaut **4444** (FYI - Ce numéro de port peut être modifié en passant le paramètre `-port` suivi du numéro de port sur lequel vous voulez exécuter le serveur).

- Ouvrez maintenant un navigateur et accédez à <http://localhost:4444/grid/console>
- Si tout fonctionne, le serveur devrait apparaître et vous verrez quelque chose comme ceci:



Ensuite, nous devons configurer des machines de noeud.

Configurer les nœuds

- Tout comme nous avons téléchargé [Selenium Server](#) for **Hub** , nous devons également le télécharger sur toutes nos machines **Node** .
- Une fois que vous avez le fichier jar Selenium-server sur la machine du noeud, accédez au répertoire où jar est téléchargé et ouvrez l'invite cmd.
- Type: `java -jar selenium-server-standalone-2.53.0.jar -role node -hub http://hubIP:4444/grid/register`
 - **hubIP** : - dans le cas où le hub et les noeuds sont exécutés sur une autre machine
 - **localhost** : - au cas où le hub et les nœuds s'exécutent sur la même machine

Comme vous pouvez le voir, le nœud est maintenant enregistré sur le concentrateur. Par défaut, le nœud démarre sur **-port 5555**, mais vous pouvez le modifier en utilisant le paramètre **-port** suivi du numéro de port.

```
➔ 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.openq
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 tout fonctionne comme prévu, vous devriez maintenant voir l'adresse IP du nœud que vous venez de démarrer et enregistré dans la vue de la console du concentrateur:



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](#)

Choses à noter

- Si nous ne spécifions pas le `seleniumProtocol`, Node sera enregistré avec Remote Control (Legacy) et Webdriver Protocol (comme indiqué dans la capture d'écran ci-dessus).
- Si le type de navigateur et le nombre d'instances ne sont pas mentionnés, Node lancera 5

instances de Firefox, 5 Instance of Chrome et 1 Instance de IE.

C'est tout ce que vous auriez besoin de faire pour une grille Selenium opérationnelle.

Configuration Json

Un exemple de configuration pour un concentrateur:

```
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"
}
```

Un exemple de configuration pour un noeud

```
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": {}
}
```

Configuration et utilisation en C

Configuration

Dans les paragraphes suivants, il y aura un exemple par navigateur pour la configuration dans Json et la configuration en C #.

Cet exemple attend de vous que tous les pilotes de votre variable de chemin et de vos navigateurs soient installés.

Microsoft Edge

Code C # pour créer un webdriver distant

```
// 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);
```

Configuration du noeud dans Json

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

Chrome

Code C # pour créer un webdriver distant

```
// 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);
```

Configuration du noeud dans Json

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

Firefox

Code C # pour créer un webdriver distant

```
// 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);
```

Configuration du noeud dans Json

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

Opéra

Code C # pour créer un webdriver distant **Ceci est pour 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);
```

Configuration du noeud dans Json

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

Le type de plate-forme peut être l'un des suivants:

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

Configuration de plusieurs navigateurs Json et C

Configuration

Dans les paragraphes suivants, il y aura un exemple par navigateur pour la configuration dans Json et la configuration en C #.

Cet exemple attend de vous que tous les navigateurs soient installés et que les pilotes soient dans votre variable de chemin

Microsoft Edge

Code C # pour créer un webdriver distant

```
// 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);
```

Configuration du noeud dans Json

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

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

Chrome

Code C # pour créer un webdriver distant

```
// 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);
```

Configuration du noeud dans Json

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

Firefox

Code C # pour créer un webdriver distant

```
// 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);
```

Configuration du noeud dans Json

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

Opéra

Code C # pour créer un webdriver distant

```
// Defining webdriver variable
```

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

Configuration du noeud dans Json

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

Le type de plate-forme peut être l'un des suivants:

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

Lire Configuration de la grille de sélénium en ligne: <https://riptutorial.com/fr/selenium-grid/topic/8702/configuration-de-la-grille-de-selenium>

Crédits

S. No	Chapitres	Contributeurs
1	Démarrer avec la grille de sélénium	Community , Paras
2	Configuration de la grille de sélénium	Paras , Thomas