



**FREE eBook**

# LEARNING selenium-grid

Free unaffiliated eBook created from  
**Stack Overflow contributors.**

#selenium-  
grid

# Table of Contents

About.....	1
<b>Chapter 1: Getting started with selenium-grid.....</b>	<b>2</b>
Remarks.....	2
Examples.....	2
What is Selenium Grid?.....	2
What is a Hub & Node?.....	2
<b>Chapter 2: Selenium Grid Configuration.....</b>	<b>4</b>
Remarks.....	4
<b>Downloads.....</b>	<b>4</b>
<b>h11.....</b>	<b>4</b>
Drivers.....	4
Browsers.....	4
Examples.....	4
Installation or Setup.....	4
Configure the Hub.....	4
Configure the Nodes.....	6
Things to Notice.....	7
Json configuration.....	8
Configuration and usage in C#.....	9
Configuration.....	9
Microsoft Edge.....	9
Chrome.....	9
Firefox.....	10
Opera.....	10
Configuration Json and C# multiple browsers.....	11
Configuration.....	11
Microsoft Edge.....	11
Chrome.....	12
Firefox.....	12
Opera.....	12



---

# About

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](mailto:info@zzzprojects.com)

---

# Chapter 1: Getting started with selenium-grid

## Remarks

This section provides an overview of what selenium-grid is, and why a developer might want to use it.

It should also mention any large subjects within selenium-grid, and link out to the related topics. Since the Documentation for selenium-grid is new, you may need to create initial versions of those related topics.

## Examples

### What is Selenium Grid?

Selenium-Grid is a configuration of *Hub & Node* which allows you to run your tests on different machines against different browser combinations in parallel. That is, running multiple tests at the same time against *different machines* running *different browsers* on *different operating systems*. In other words Selenium Grid supports running the tests in **Distributed Environment**.

### When to use it

- **To run your tests against multiple browsers, multiple versions of browser, and browsers running on different operating systems.** This will ensure that the application you are testing is fully compatible with a wide range of browser-O.S combinations.
- **To reduce the time it takes for the test suite to complete a test pass.** Lets say you setup your Grid to run 8 tests at a time, your execution would finish 8 times faster as compared to your normal run.

Selenium-Grid is used to speed up the execution of a test by using multiple machines to run tests in parallel.

### What is a Hub & Node?

#### The Hub

- The *Hub* is the main engine/central point of the entire configuration, point where all the nodes are connected.
- *Hub* should run only on a single machine.
- There should only be 1 hub running where all the tests are loaded.
- Tests will be run on the machines where hub is running, but you can see the browsers on the node machines.

#### The Nodes

- Nodes are the instances (machines) which will execute the tests that are loaded on the hub.

- There are no limitations on Node machines, a user can setup n number of Nodes.
- Nodes can be launched on different machines with different OS and browser combinations.
- Machines running the nodes can be of different/same configurations as of Hub Machine.

Read **Getting started with selenium-grid** online: <https://riptutorial.com/selenium-grid/topic/8701/getting-started-with-selenium-grid>

---

# Chapter 2: Selenium Grid Configuration

## Remarks

---

## Downloads

This chapter contains useful downloads like the webdrivers and link to browsers

---

## Drivers

Place all the drivers in your path variable

- [Chrome driver](#)
- [FireFox driver](#)
- [Microsoft Edge driver](#)
- [Opera driver](#)

## Browsers

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

## Examples

### Installation or Setup

Before setting up a Selenium grid you need to make sure you have *Java* installed and configured in your computer's environment path.

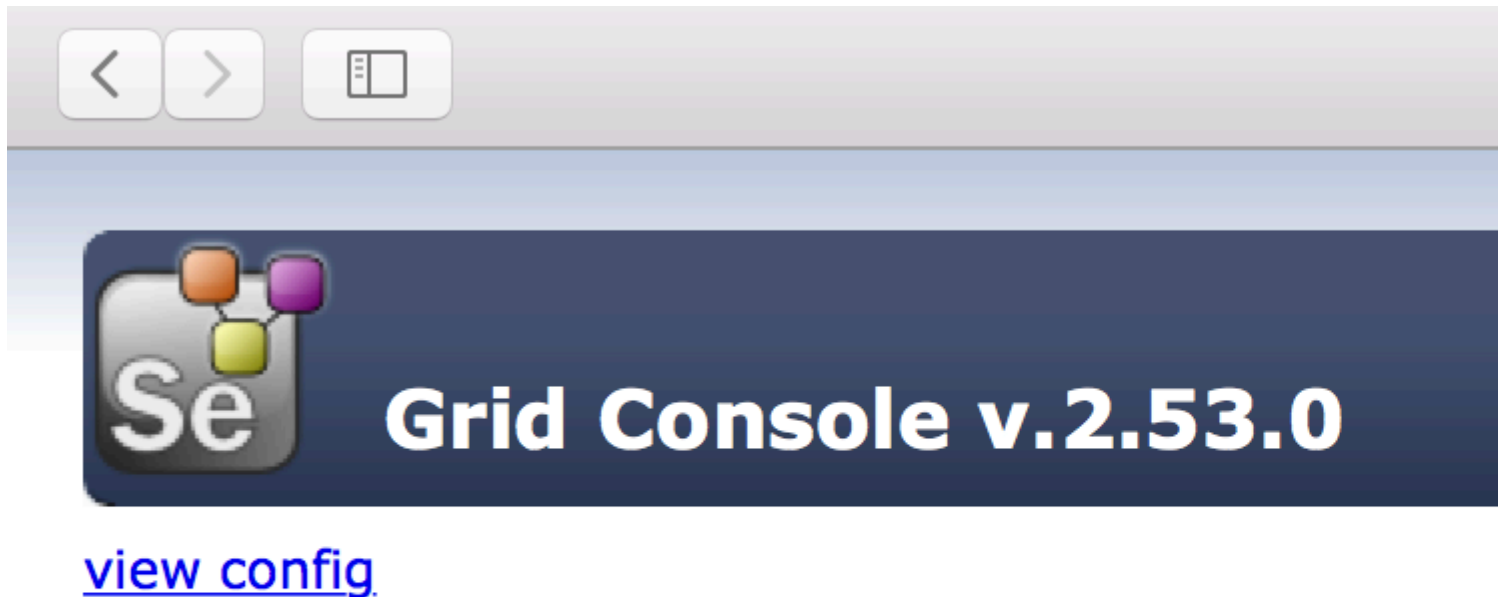
## Configure the Hub

- Download latest stable [Selenium Server](#) version.
- Start the command prompt and navigate to the location in which you placed the Selenium server jar file.
- Type: (FYI: your version number may be different than mine) `java -jar selenium-server-standalone-2.53.0.jar -role hub`
- It should now look something like this:

```
→ 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
```

Basically what happened is selenium webserver started and is now listening on a port - in this case default **4444** (FYI - This port number can be changed by passing the *-port* parameter followed by the port number on which you want to run the server).

- Now open a browser and navigate to <http://localhost:4444/grid/console>
- If everything is working, server should come up and you would see something like this:



Next, we need to setup some node machines.



## Configure the Nodes

- Just like we downloaded [Selenium Server](#) for **Hub**, we also need to download it on all our **Node** machines.
- Once you have the Selenium-server jar file on the node machine, navigate to the directory where jar is downloaded and open up cmd prompt.
- Type: `java -jar selenium-server-standalone-2.53.0.jar -role node -hub http://hubIP:4444/grid/register`
  - **hubIP** :- in case the hub and nodes are running on a different machine
  - **localhost** :- in case the hub and nodes are running on same machine

As you can see the node is now registered to the hub, by default node starts on **-port 5555** but you can change the same by using **-port** parameter followed by port number.

```
→ 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
```

If everything works as expected you should now see the IP address of the node you just started and registered in the hub console view:



# 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](#)

## Things to Notice

- If we don't specify the `seleniumProtocol`, Node will be registered with both Remote Control (Legacy) and Webdriver Protocol (as seen in the screenshot above).
- If the browsers type and number of instances aren't mentioned, Node will launch 5 instance

of Firefox, 5 Instance of Chrome and 1 Instance of IE driver.

That's all you would be needing to do for an up and running Selenium Grid.

## Json configuration

An example configuration for a 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"
}
```

An example configuration for a node

```
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 and useage in C#

### Configuration

In the following paragraphs there will be an example per browser for the configuration in Json and setup in C#.

This example expects you to have all the drivers in your path variable and browsers installed.

#### Microsoft Edge

##### C# code to create a remote webdriver

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

##### Node configuration in Json

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

#### Chrome

##### C# code to create a remote webdriver

```
// 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);
```

## Node configuration in Json

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

## Firefox

### C# code to create a remote webdriver

```
// 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);
```

## Node configuration in Json

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

## Opera

### C# code to create a remote webdriver **This is for 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);
```

## Node configuration in Json

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

Platform type can be one of the following:

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

## Configuration Json and C# multiple browsers

### Configuration

In the following paragraphs there will be an example per browser for the configuration in Json and setup in C#.

This example expects you to have all the browsers installed and the drivers in your path variable

### Microsoft Edge

C# code to create a remote webdriver

```
// 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);
```

## Node configuration in Json

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

## Chrome

### C# code to create a remote webdriver

```
// 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);
```

### Node configuration in Json

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

## Firefox

### C# code to create a remote webdriver

```
// 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);
```

### Node configuration in Json

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

## Opera

### C# code to create a remote webdriver

```
// 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);
```

## Node configuration in Json

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

Platform type can be one of the following:

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

Read Selenium Grid Configuration online: <https://riptutorial.com/selenium-grid/topic/8702/selenium-grid-configuration>



---

# Credits

S. No	Chapters	Contributors
1	Getting started with selenium-grid	<a href="#">Community</a> , <a href="#">Paras</a>
2	Selenium Grid Configuration	<a href="#">Paras</a> , <a href="#">Thomas</a>