

5. Selenium Grid

WHAT IS SELENIUM GRID?
ARCHITECTURE
WORKING WITH GRID
CONFIGURING THE HUB
CONFIGURING THE NODES
CODES

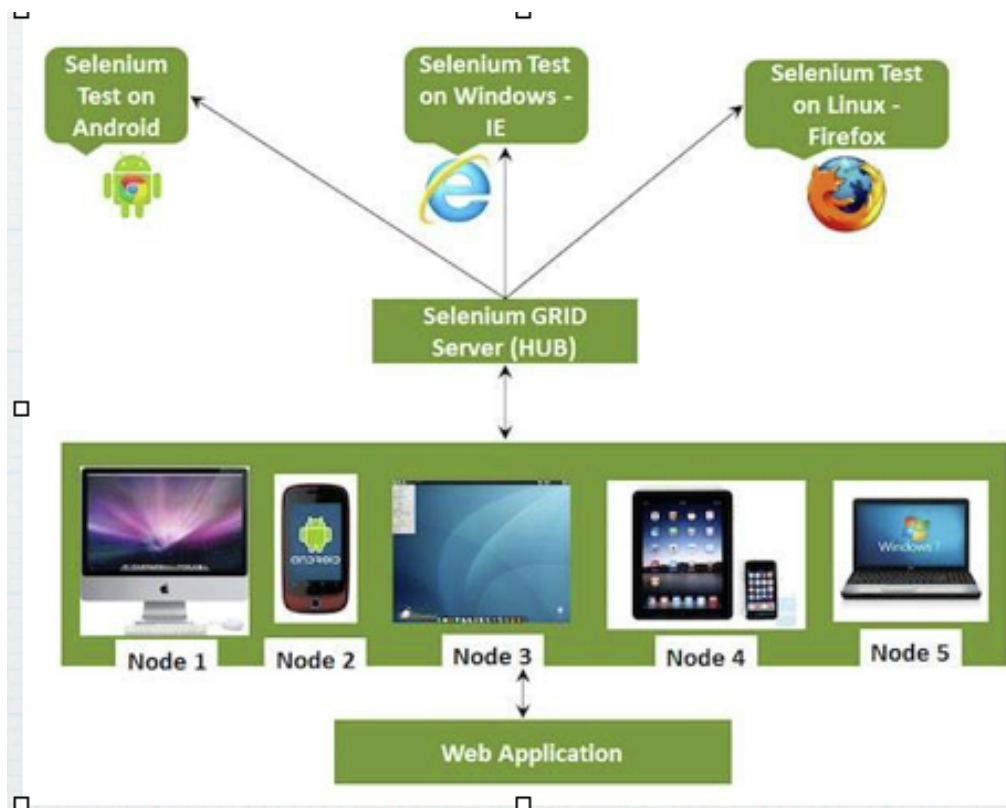
What is Selenium Grid?

- Selenium Grid is a part of the Selenium Suite that specializes on running multiple tests across different browsers, operating systems, and machines in parallel.
- Selenium Grid allows us to execute multiple instances of WebDriver or Selenium Remote Control tests in parallel which uses the same code base, hence the code need NOT be present on the system they execute. The selenium-server-standalone package includes Hub, WebDriver, and Selenium RC to execute the scripts in grid.

What is Selenium Grid?

- Selenium Grid has a Hub and a Node.
- **Hub** - The hub can also be understood as a server which acts as the central point where the tests would be triggered. A Selenium Grid has only one Hub and it is launched on a single machine once.
- **Node** - Nodes are the Selenium instances that are attached to the Hub which execute the tests. There can be one or more nodes in a grid which can be of any OS and can contain any of the Selenium supported browsers.

Architecture



Working with Grid

- In order to work with the Grid, we need to follow certain protocols. Listen below are the major steps involved in this process:
- Configuring the Hub
- Configuring the Nodes
- Develop the Script and Prepare the XML File
- Test Execution
- Result Analysis

Configuring the Hub

Step 1 : Download the latest Selenium Server standalone JAR file from <http://docs.seleniumhq.org/download/>. Download it by clicking on the version as shown below



SeleniumHQ
Browser Automation

edit this page search selenium: Go

Projects Download Documentation Support About

Selenium Downloads

- Previous Releases
- Source Code
- Maven Information

Donate to Selenium

with PayPal

Donate

through sponsorship

You can sponsor the Selenium project if you'd like some public recognition of your generous contribution.

Selenium Sponsors

See who supports the Selenium project.

BrowserStack

SAUCE LABS

experitest
Selenium for Mobile

SYNTHETICS
New Relic

Downloads

Below is where you can find the latest releases of all the Selenium components. You can also find a list of [previous releases](#), [source code](#), and additional information for [Maven users](#) (Maven is a popular Java build tool).

Selenium Standalone Server

The Selenium Server is needed in order to run Remote Selenium WebDriver. Selenium 3.X is no longer capable of running Selenium RC directly, rather it does it through emulation and the [WebDriverBackedSelenium](#) interface.

Download version **3.0.1**

To run Selenium tests exported from IDE, use the [Selenium Html Runner](#).

To use the Selenium Server in a Grid configuration [see the wiki page](#).

The Internet Explorer Driver Server

This is required if you want to make use of the latest and greatest features of the WebDriver InternetExplorerDriver. Please make sure that this is available on your \$PATH (or %PATH% on Windows) in order for the IE Driver to work as expected.

Download version 2.53.1 for (recommended) [32 bit Windows IE](#) or [64 bit Windows IE](#)

[CHANGELOG](#)

Selenium Client & WebDriver Language Bindings

In order to create scripts that interact with the Selenium Server (Selenium RC, Selenium Remote WebDriver) or create local Selenium WebDriver scripts, you need to make use of language-specific client drivers. These languages include both 1.x and 2.x style clients.

While language bindings for [other languages exist](#), these are the core ones that are supported by the main project hosted on google code.

Language	Client Version	Release Date	Download	Change log	Javadoc
Java	3.0.1	2016-10-18	Download	Change log	Javadoc
C#	3.0.0	2016-10-13	Download	Change log	API docs
Ruby	3.0.0	2016-10-13	Download	Change log	API docs
Python	Selenium 3.0.0.b2	2016-08-03	Download	Change log	API docs
Javascript (Node)	3.0.0-beta-2	2016-08-07	Download	Change log	API docs

C# NuGet

NuGet latest release is 3.0.0, Released on 2016-10-13

Step 2 : Start the Hub by launching the Selenium Server using the following command. Now we will use the port '4444' to start the hub.

Note : Ensure that there are no other applications that are running on port# 4444.

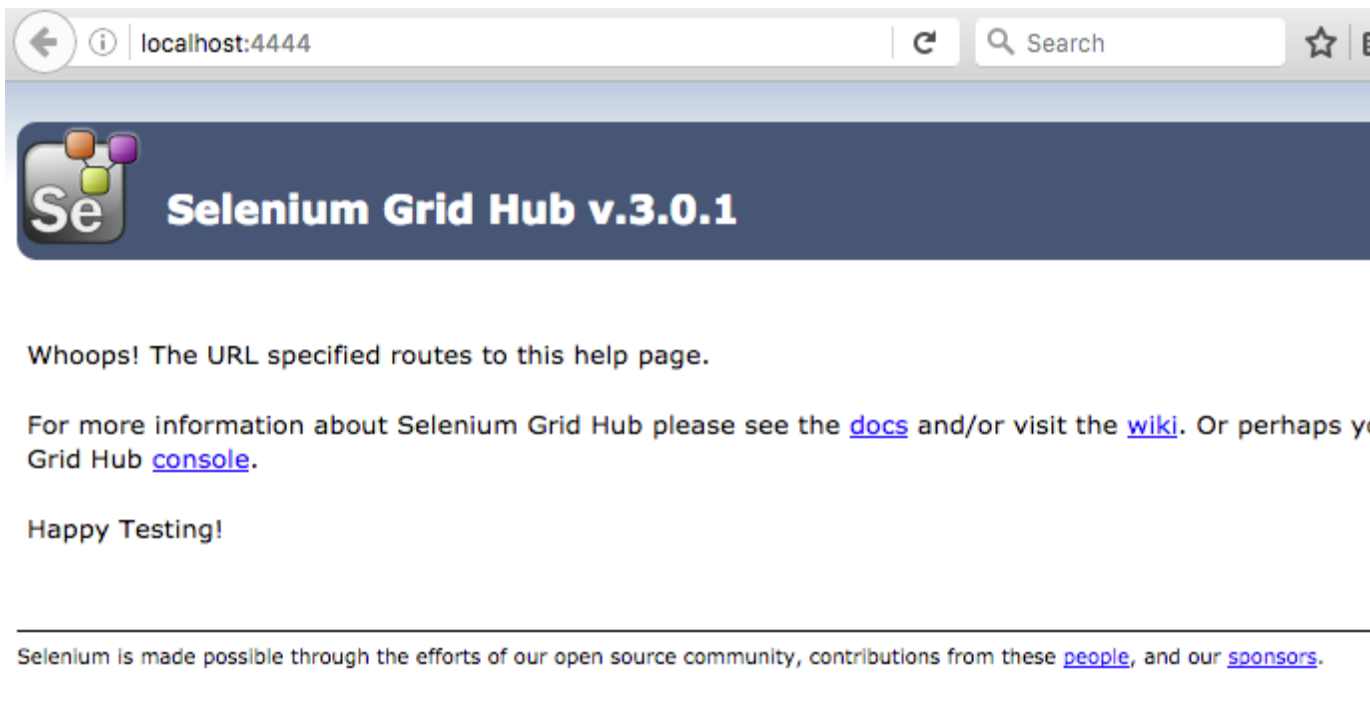
- cd to the folder which contains selenium-server-standalone-3.0.1.jar

- run terminal

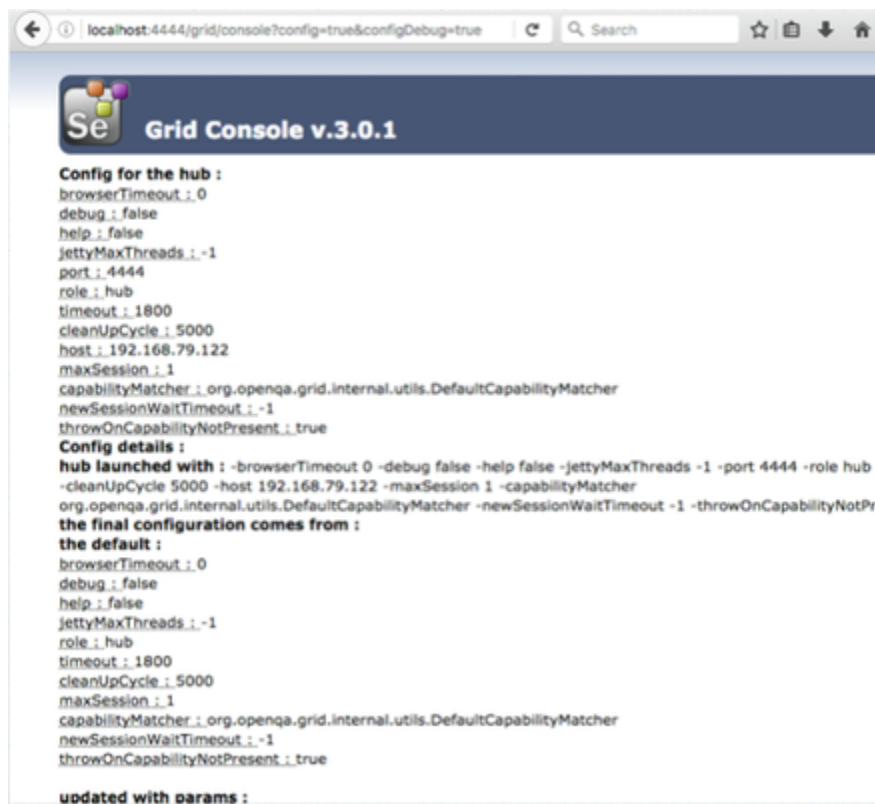
java -jar selenium-server-standalone-3.0.1.jar -port 4444 -role hub

- cd to the folder which contains selenium-server-standalone-3.0.1.jar

Step 3 : Now open the browser and navigate to the URL <http://localhost:4444> from the Hub (The system where you have executed Step#2).



Step 4 : Now click on the 'console' link and click 'view config'. The config of the hub would be displayed as follows. As of now, we haven't got any nodes, hence we will not be able to see the details.



Configuring the Nodes

Step 1 : Logon to the node (where you would like to execute the scripts) and place the 'selenium-server-standalone-3.0.1.jar' in a folder. We need to point to the selenium-server-standalone JAR while launching the nodes.

- cd to the folder which contains selenium-server-standalone-3.0.1.jar

- open other terminal

java -jar selenium-server-standalone-3.0.1.jar -port 5555 -role node -hub <http://localhost:4444/grid/register> -browser "browserName=firefox, maxInstances=10, platform=ANY, seleniumProtocol=WebDriver"

Step 2 : After executing the command, come back to the Hub. Navigate to the URL - <http://localhost:4444> and the Hub would now display the node attached to it.



- Chrome

MAC: java -Dwebdriver.chrome.driver=/Users/nguyenthithongthuy/Documents/Teaching/chromedriver -jar selenium-server-standalone-3.0.1.jar -role webdriver -hub <http://localhost:4444/grid/register> -browser browserName=chrome,platform=ANY -port 5557

Windows: java -Dwebdriver.chrome.driver=D:\chromedriver.exe -jar D:\JAR\selenium-server-standalone-3.0.1.jar -role webdriver -hub <http://10.30.217.157:4444/grid/register> -browser browserName=chrome,platform=WINDOVS -port 5557



- IE

MAC: java -Dwebdriver.ie.driver=D:\IEDriverServer.exe -jar D:\JAR\selenium-server-standalone-3.0.1.jar -role webdriver -hub <http://localhost:4444/grid/register> -browser browserName=ie,platform=ANY -port 5558

Windows: java -Dwebdriver.ie.driver=D:\IEDriverServer.exe -jar D:\JAR\selenium-server-standalone-3.0.1.jar -role webdriver -hub <http://10.30.217.157:4444/grid/register> -browser browserName=ie,platform=WINDOVS -port 5558

Codes

```
if (browser.equalsIgnoreCase("firefox"))  
  
    {  
  
        System.out.println(" Executing on FireFox");  
  
        String Node = "http://10.112.66.52:5555/wd/hub";  
  
        DesiredCapabilities cap = DesiredCapabilities.firefox();  
  
        cap.setBrowserName("firefox");  
  
  
        driver = new RemoteWebDriver(new URL(Node), cap);  
  
        // Puts an Implicit wait, Will wait for 10 seconds before  
        throwing exception  
  
        driver.manage().timeouts().implicitlyWait(10,  
TimeUnit.SECONDS);  
  
  
        // Launch website  
  
        driver.navigate().to(URL);  
  
        driver.manage().window().maximize();  
  
    }
```

```
else if (browser.equalsIgnoreCase("chrome"))  
  
    {  
  
        System.out.println(" Executing on CHROME");  
  
        DesiredCapabilities cap = DesiredCapabilities.chrome();  
  
        cap.setBrowserName("chrome");  
  
        String Node = "http://10.112.66.52:5557/wd/hub";  
  
        driver = new RemoteWebDriver(new URL(Node), cap);  
  
        driver.manage().timeouts().implicitlyWait(10,  
TimeUnit.SECONDS);  
  
  
        // Launch website  
  
        driver.navigate().to(URL);  
  
        driver.manage().window().maximize();  
  
    }
```

```

else if (browser.equalsIgnoreCase("ie"))
{
    System.out.println(" Executing on IE");

    DesiredCapabilities cap = DesiredCapabilities.chrome();

    cap.setBrowserName("ie");

    String Node = "http://10.112.66.52:5558/wd/hub";

    driver = new RemoteWebDriver(new URL(Node), cap);

    driver.manage().timeouts().implicitlyWait(10,
TimeUnit.SECONDS);

    // Launch website

    driver.navigate().to(URL);

    driver.manage().window().maximize();
}

else
{
    throw new IllegalArgumentException("The Browser Type is
Undefined");
}

```

Preference

https://www.tutorialspoint.com/selenium/selenium_grids.htm

<http://www.seleniumeasy.com/selenium-tutorials/how-to-configure-selenium-grid>

<http://www.guru99.com/introduction-to-selenium-grid.html>