Selenium Grid

Selenium Grid is a proxy server part of the Selenium suite. It makes running tests in parallel on multiple machines easy and centrally manages different browser versions and browser configurations.

Selenium 4 supports three grid types:

* Standalone Mode
* Classical Grid (Hub and Node)
* Fully Distributed (Router, Distributor, Session, and Node)

# Advantages of Grid 4

* Provides a wide range of [programming language support](https://www.selenium.dev/ecosystem/).
* Supports multiple browsers and multiple operating systems just like [Selenium Webdriver](https://www.browserstack.com/guide/selenium-webdriver-tutorial).
* It is free and open-source.
* It supports [Parallel Test Execution](https://www.browserstack.com/guide/parallel-testing-with-selenium) (Local and Cloud-Based).

# Downloading Grid

<https://github.com/SeleniumHQ/selenium/releases/download/selenium-4.23.0/selenium-server-4.23.1.jar>

By default, the server will be listening on **http://localhost:4444** port

# Setting up grid in standalone mode

1. Download the above-mentioned jar file.
2. Go to the location where the jar file is downloaded
3. Execute the below command

java -jar selenium-server-4.0.0-alpha-7.jar standalone

Standalone server starts at URL <http://localhost:4444>

1. Add the below code in the test script file

ChromeOptions opt = **new** ChromeOptions();

WebDriver driver1 = **new** RemoteWebDriver(**new** URL("https:\\localhost:4444"),opt);

# Setting up grid in Hub-Node

Hub and Node is the most used role because it allows to:

* Combine different machines in a single Grid
  + Machines with different operating systems and/or browser versions, for example
* Have a single entry point to run WebDriver tests in different environments

## Hub and Node on the same machine

### To start Hub

java -jar selenium-server-4.7.2.jar hub

### To start the node

java -jar selenium-server-4.7.2.jar node

Multiple nodes can be set up using below command and by giving different port numbers.

java -jar selenium-server-<version>.jar node --port 5555

java -jar selenium-server-<version>.jar node --port 6666

## Hub and Node Different machines

### Hubs

* The hub is the central point where you load your tests into.
* There should only be one hub in a grid.
* The machine containing the hub is where the tests will be run, but you will see the browser being automated on the node.

### Nodes

* Nodes are the Selenium instances that will execute the tests that you loaded on the hub.
* There can be one or more nodes in a grid.
* Nodes can be launched on multiple machines with different platforms and browsers.
* The machines running the nodes need not be the same platform as that of the hub.

Hub and Nodes talk to each other via HTTP and the Event Bus

**To set up Hub machine -** java -jar selenium-server-4.7.2.jar hub

After starting the Hub with default ports, the **--hub** flag can be used to register the Node

java -jar selenium-server-<version>.jar node --hub http://<hub-ip>:4444

Now by adding the below Chrome options, we can run test on node.

ChromeOptions opt = **new** ChromeOptions();

WebDriver driver1 = **new** RemoteWebDriver(**new** URL("https:\\localhost:4444"),opt);

Chrome Options

ChromeOptions options = new ChromeOptions()

options.addArgument("start-maximized");

ChromeDriver driver = new ChromeDriver(options);

Below are the list of available and most commonly used arguments for ChromeOptions class

* **start-maximized**: Opens Chrome in maximize mode
* **incognito:**Opens Chrome in incognito mode
* **headless:** Opens Chrome in headless mode
* **disable-extensions**: Disables existing extensions on Chrome browser
* **disable-popup-blocking**: Disables pop-ups displayed on Chrome browser
* **make-default-browser:** Makes Chrome default browser
* **version**: Prints chrome browser version
* **disable-infobars:** Prevents Chrome from displaying the notification ‘Chrome is being controlled by automated software