

13. Grid Configuration Using JSON

1. Configuring the grid hub using JSON

- a. Create JSON file for the hub which looks like

```
{
  "port": 4444,
  "newSessionWaitTimeout": -1,
  "servlets": [],
  "withoutServlets": [],
  "custom": {},
  "capabilityMatcher": "org.openqa.grid.internal.utils.DefaultCapabilityMatcher",
  "throwOnCapabilityNotPresent": true,
  "cleanUpCycle": 5000,
  "role": "hub",
  "debug": false,
  "browserTimeout": 0,
  "timeout": 1800
}
```

- b. Save it in a folder with a valid name (example: myhub) in which we have saved Selenium standalone Server jar file
- c. Go to command prompt
- d. Navigate to folder structure where you have saved the Selenium standalone Server jar file
- e. Type the below command in command prompt

Java -jar selenium-server-standalone-3.141.59.jar -role hub -hubConfig myhub.json and click on **Enter** button, which looks like

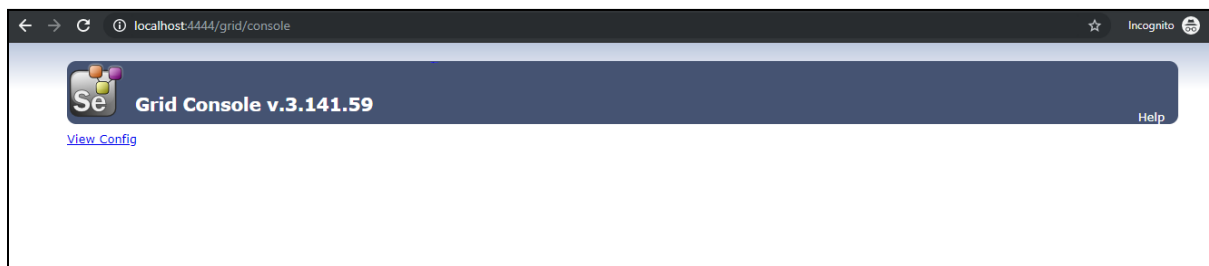
```
Command Prompt - java -jar selenium-server-standalone-3.141.59.jar -role hub -hubConfig myhub.json
Microsoft Windows [Version 10.0.17134.950]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\User>d:

D:\>cd Selenium Grid

D:\Selenium Grid>java -jar selenium-server-standalone-3.141.59.jar -role hub -hubConfig myhub.json
15:44:15.021 INFO [GridLauncherV3.parse] - Selenium server version: 3.141.59, revision: e82be7d358
15:44:15.157 INFO [GridLauncherV3.lambda$buildLaunchers$5] - Launching Selenium Grid hub on port 4444
2019-08-20 15:44:15.581:INFO::main: Logging initialized @929ms to org.seleniumhq.jetty9.util.log.StdErrLog
15:44:16.217 INFO [Hub.start] - Selenium Grid hub is up and running
15:44:16.218 INFO [Hub.start] - Nodes should register to http://192.168.1.248:4444/grid/register/
15:44:16.218 INFO [Hub.start] - Clients should connect to http://192.168.1.248:4444/wd/hub
15:49:28.797 INFO [DefaultGridRegistry.add] - Registered a node http://192.168.1.248:5555
```

- f. Open the Chrome browser
- g. Enter URL as 'http://localhost:4444/grid/console' and click on enter
- h. Grid console page is loaded as below



2. Configuring the grid nodes using JSON

- a. Once the Selenium Grid Hub using JSON is configured, the next step is to configure Selenium Grid nodes using JSON.
- b. Create JSON file for node, which looks like:

```

{
  "capabilities": [
    {
      "browserName": "firefox",
      "maxInstances": 5,
      "seleniumProtocol": "WebDriver"
    },
    {
      "browserName": "chrome",
      "maxInstances": 5,
      "seleniumProtocol": "WebDriver"
    }
  ],
  "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": {}
}

```

- c. Save it in a folder with a valid name (example: mynode) in which we have saved Selenium standalone Server jar file
- d. Open the new command prompt
- e. Navigate to the folder structure where you have saved the Selenium standalone server jar file
- f. Type the below command in command prompt

```

java -Dwebdriver.gecko.driver="geckodriver.exe" -  

Dwebdriver.chrome.driver="chromedriver.exe" -jar  

selenium-server-standalone-3.141.59.jar -role node -nodeConfig  

mynodes.json

```

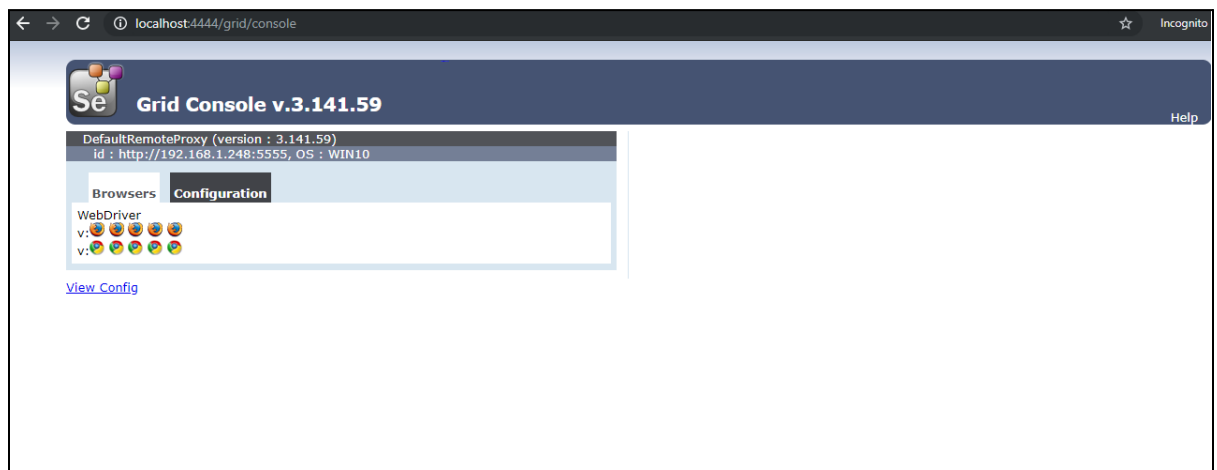
and click on **Enter** button, which looks like

```
Command Prompt - java -Dwebdriver.gecko.driver="geckodriver.exe" -Dwebdriver.chrome.driver="chromedriver.exe" -jar selenium-server-standalone...
Microsoft Windows [Version 10.0.17134.950]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\User>cd Selenium Grid

D:\Selenium Grid>java -Dwebdriver.gecko.driver="geckodriver.exe" -Dwebdriver.chrome.driver="chromedriver.exe" -jar selenium-server-standalone-3.141.59.jar -role node -nodeConfig mynodes.json
16:05:36.650 INFO [GridLauncherV3.parse] - Selenium server version: 3.141.59, revision: e82be7d358
16:05:36.809 INFO [GridLauncherV3.lambda$buildLaunchers$7] - Launching a Selenium Grid node on port 5555
2019-08-20 16:05:37.511:INFO::main: Logging initialized @1177ms to org.seleniumhq.jetty9.util.log.StdErrLog
16:05:37.856 INFO [WebDriverServlet.<init>] - Initialising WebDriverServlet
16:05:37.959 INFO [SeleniumServer.boot] - Selenium Server is up and running on port 5555
16:05:37.959 INFO [GridLauncherV3.lambda$buildLaunchers$7] - Selenium Grid node is up and ready to register to the hub
16:05:38.225 INFO [SelfRegisteringRemote$1.run] - Starting auto registration thread. Will try to register every 5000 ms.
16:05:38.769 INFO [SelfRegisteringRemote.registerToHub] - Registering the node to the hub: http://localhost:4444/grid/register
16:05:38.992 INFO [SelfRegisteringRemote.registerToHub] - The node is registered to the hub and ready to use
```

- g. Open the browser
- h. Enter URL as **http://localhost:4444/grid/console** and click on **Enter**
- i. Grid console page is loaded, which shows **Browsers** by default



- j. Click on **Configuration** which shows Configuration details



Pushing the code to your GitHub repositories : -

- Open your folder where the Project . And then click the right button to open the git bash command prompt.
- Before that, open the github and create a new repository.
- Initialize your repository using the following command:

git init

- Add all the files to your git repository using the following command:

git add .

- To check the status of the repository use the below command:

git status

- Commit the changes using the following command:

git commit . -m "Changes have been committed."

- To add the files to the repository use the (URL) from the github and use the command;

git remote add origin <url>

- Push the files to the folder you initially created using the following command:

git push origin master.