Extend Selenium Grid in Test Automation Projects

Hello everybody, **Selenium Grid** is used to speed up the execution of a test pass by using multiple machines to run tests in parallel. It routes the test suites to available browsers and decreases the test execution time. Today, we are going to describe how to **extend Selenium Grid** for your needs. It's going to be Extending Selenium Grid 101 course. 🙂 Let's learn how to extend Selenium Grid!

Creating Your Extension Projects

- Create a Quickstart Maven Project. The most important part is the **GroupId** and **ArtifactId**. They are going to be crucial in next steps. Our GroupId is "**com.swtestacademy**" and ArtifactId is "**ExtendedGrid**".
- Add "**Selenium Standalone Server**" dependency into your pom.xml

```XHTML
<dependency>
 <groupId>org.seleniumhq.selenium</groupI
 <artifactId>selenium-server</artifactId>
 <version>3.12.0</version>
</dependency>
```

You're ready to extend Selenium Grid. Let's start coding.

Code Implementation

Create a java class and extend it by using **DeafaultRemoteProxy** class and implement **TestSessionListener** Interface.

```Java
public class ExtendedProxy extends Defaul
```

Then you'll be able to override some methods. Those methods are:

- **BeforeSession** (It's trigger before a session is created)
- **AfterSession** (It's trigger after a session is finished)
- **BeforeCommand** (It's trigger before a command is executed)
- **AfterCommand** (It's trigger after a command is executed)

Let's do some basic stuff with them as below in this class.

```Java
@Override
public void beforeCommand(TestSession se
                          HttpServletResp
   System.out.println("****** SWTESTACAI
   System.out.println("Selenium Extendir
   System.out.println("Method " +request
   System.out.println("getRequestURI " +
   System.out.println("Session "+sessior

}

@Override
public void afterCommand(TestSession ses
                          HttpServletResp
   System.out.println("****** SWTESTACAI
   System.out.println("Selenium Extendir
}

@Override
public void beforeSession(TestSession se
   System.out.println("****** SWTESTACAI
   System.out.println("Selenium Extendir
```

```
}

@Override
public void afterSession(TestSession ses
   System.out.println("****** SWTESTACAI
   System.out.println("Selenium Extendir
}
```

Build Your Grid

In order to build your Selenium Grid and export a jar file. You need to do some modifications into your **pom.xml**. You're also going to use different maven command while building your project.

Add "**maven assembly plugin**" into your pom.xml's plugins section. All you need to change is the value inside **MainClass** tag according to your package names. With this plugin, all the dependencies will be included in your extended grid jar files.

```xml
<plugin>
 <groupId>org.apache.maven.plugins</grou
 <artifactId>maven-assembly-plugin</arti
 <executions>
   <execution>
     <id>create-my-bundle</id>
     <phase>package</phase>
     <goals>
       <goal>single</goal>
     </goals>
     <configuration>
       <archive>
         <manifest>
           <mainClass>com.swtestacademy.
           </mainClass>
         </manifest>
         <manifestEntries>
           <Class-Path>.</Class-Path>
         </manifestEntries>
       </archive>
       <descriptorRefs>
         <descriptorRef>jar-with-depende
       </descriptorRefs>
     </configuration>
   </execution>
 </executions>
</plugin>
```

Add below plugin into plugins section also. By this way, you specify the main class that will be executed when Grid is launched. This main class is "**org.openqa.grid.selenium.GridLauncherV3**". You shouldn't change this value.

```xml
<plugin>
 <artifactId>maven-assembly-plugin</arti
 <version>2.5</version>
 <configuration>
   <descriptorRefs>
     <descriptorRef>jar-with-dependencie
   </descriptorRefs>
   <archive>
     <manifest>
       <mainClass>org.openqa.grid.seleni
       <addDefaultImplementationEntries>
     </manifest>
   </archive>
 </configuration>
 <executions>
   <execution>
     <id>make-assembly</id> <!-- this is
     <phase>package</phase> <!-- bind to
     <goals>
       <goal>single</goal>
     </goals>
   </execution>
 </executions>
</plugin>
<plugin>
 <groupId>org.codehaus.mojo</groupId>
 <artifactId>exec-maven-plugin</artifact
```

```
<version>1.2.1</version>
<executions>
  <execution>
    <goals>
      <goal>java</goal>
    </goals>
  </execution>
</executions>
<configuration>
  <mainClass>org.openqa.grid.selenium.G
</configuration>
</plugin>
```

Build Your Package

Run below maven command to build your jar file with all the dependencies.

```shell
mvn clean compile assembly:single
```

Now you have a target folder like these with **extenededGrid-1.0-SNAPSHOT-jar-with-dependencies.jar** file. Copy this jar file into another folder where you'll run your Grid.

Start You Extended Grid as a HUB

Run your ExtendedGrid with the new jar file. Actually, there's no difference compared to the standard Selenium Grid. Only name of the jar file is changed.

```shell
java -jar extenededGrid-1.0-SNAPSHOT-jar-
```

Then your Grid will run as HUB. You should be able to reach SeleniumGrid console by browsing http://localhost:4444/grid/console.

Connect Nodes to Extended Hub

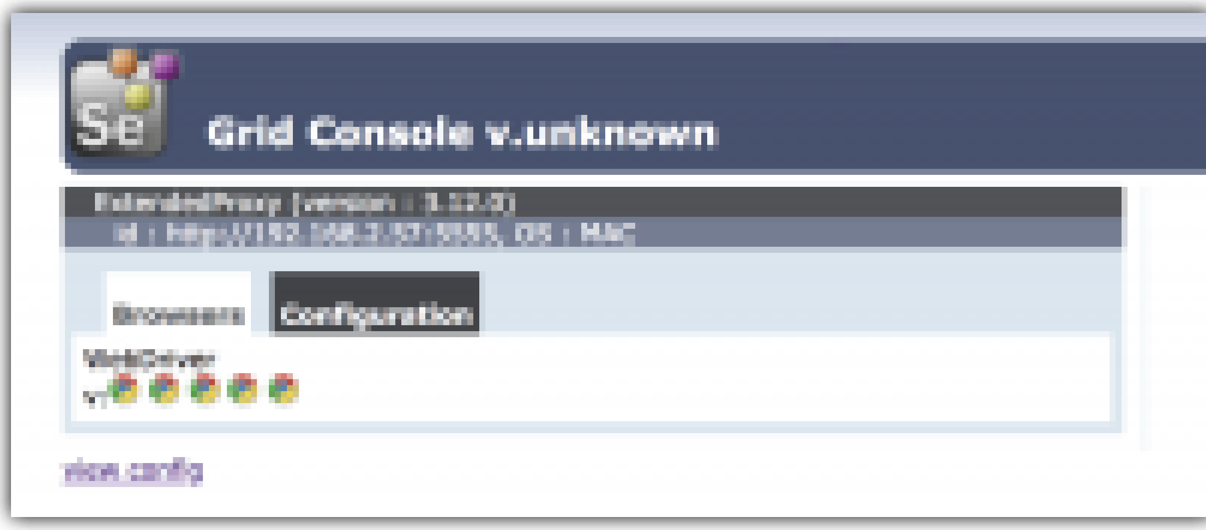
You need to create **node.json** file for your nodes. Let's create a basic one. This node will be a Chrome Browser node with 5 instance. **The most important part is to change the proxy value with the extended java class**. By this way, all your command will be proxied by this class. In case you keep it as the default which is "**org.openqa.grid.selenium.proxy.DefaultRemoteProxy**"your extended class will not work.

```java
{
  "capabilities":
  [
    {
      "browserName": "chrome",
      "maxInstances": 5,
      "seleniumProtocol": "WebDriver"
    }
  ],
  "proxy": "com.swtestacademy.ExtendedPr
  "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": {}
}
```

Save that as node.json into a folder. Run your Grid nodes

```shell
java -Dwebdriver.chrome.driver="chromedri
```

Your Nodes should be ready if you go into your Grid console, you will see below screenshot.



Try Your Extended Selenium Grid

Create a simple Selenium project with some click, sendKeys events and run it. If you see your sysout command on the console, it means your extended grid works like a charm.

```java
11:50:37.279 INFO [TestSlot.getNewSessic
****** SWTESTACADEMY *****
Selenium Extending Grid - Before Sessior
****** SWTESTACADEMY *****
Selenium Extending Grid - Before Commanc
Method POST
getRequestURI /wd/hub/session
Session 026aad5c-cc6c-4c12-a23e-b7a8c7f9
****** SWTESTACADEMY *****
Selenium Extending Grid - After Command
****** SWTESTACADEMY *****
Selenium Extending Grid - Before Commanc
Method POST
getRequestURI /wd/hub/session/3ea3677c0f
Session ext. key 3ea3677c0fa99c9c41a96a9
****** SWTESTACADEMY *****
Selenium Extending Grid - After Command
****** SWTESTACADEMY *****
Selenium Extending Grid - Before Commanc
Method GET
getRequestURI /wd/hub/session/3ea3677c0f
Session ext. key 3ea3677c0fa99c9c41a96a9
****** SWTESTACADEMY *****
Selenium Extending Grid - After Command
****** SWTESTACADEMY *****
Selenium Extending Grid - Before Commanc
Method POST
getRequestURI /wd/hub/session/3ea3677c0f
Session ext. key 3ea3677c0fa99c9c41a96a9
****** SWTESTACADEMY *****
Selenium Extending Grid - After Command
****** SWTESTACADEMY *****
Selenium Extending Grid - Before Commanc
Method POST
getRequestURI /wd/hub/session/3ea3677c0f
Session ext. key 3ea3677c0fa99c9c41a96a9
****** SWTESTACADEMY *****
Selenium Extending Grid - After Command
****** SWTESTACADEMY *****
Selenium Extending Grid - Before Commanc
Method POST
getRequestURI /wd/hub/session/3ea3677c0f
Session ext. key 3ea3677c0fa99c9c41a96a9
****** SWTESTACADEMY *****
Selenium Extending Grid - After Command
****** SWTESTACADEMY *****
Selenium Extending Grid - Before Commanc
Method POST
getRequestURI /wd/hub/session/3ea3677c0f
Session ext. key 3ea3677c0fa99c9c41a96a9
****** SWTESTACADEMY *****
Selenium Extending Grid - After Command
```

We'll add some functionalities into our Extended Grid in the next tutorials.