References - <https://github.com/temyers/cucumber-jvm-parallel-plugin>

<plugin> <groupId>com.github.temyers</groupId> <artifactId>cucumber-jvm-parallel-plugin</artifactId> <version>2.1.0</version> <executions> <execution> <id>generateRunners</id> <phase>generate-test-sources</phase> <goals> <goal>generateRunners</goal> </goals> <configuration> <!-- Mandatory --> <!-- comma separated list of package names to scan for glue code --> <glue>foo, bar</glue> <!-- These are optional, with the default values --> <!-- Where to output the generated tests --> <outputDirectory>${project.build.directory}/generated-test-sources/cucumber</outputDirectory> <!-- The directory, which must be in the root of the runtime classpath, containing your feature files. --> <featuresDirectory>src/test/resources/features/</featuresDirectory> <!-- Directory where the cucumber report files shall be written --> <cucumberOutputDir>target/cucumber-parallel</cucumberOutputDir> <!-- comma separated list of output formats --> <format>json</format> <!-- CucumberOptions.strict property --> <strict>true</strict> <!-- CucumberOptions.monochrome property --> <monochrome>true</monochrome> <!-- The tags to run, maps to CucumberOptions.tags property --> <tags></tags> <!-- If set to true, only feature files containing the required tags shall be generated. --> <filterFeaturesByTags>false</filterFeaturesByTags> <!-- Generate TestNG runners instead of JUnit ones. --> <useTestNG>false</useTestNG> <!-- The naming scheme to use for the generated test classes. One of 'simple' or 'feature-title' --> <namingScheme>simple</namingScheme> <!-- The class naming pattern to use. Only required/used if naming scheme is 'pattern'.--> <namingPattern>Parallel{c}IT</namingPattern> <!-- One of [SCENARIO, FEATURE]. SCENARIO generates one runner per scenario. FEATURE generates a runner per feature. --> <parallelScheme>SCENARIO</parallelScheme> <!-- Specify a custom template for the generated sources (this is a path relative to the project base directory) --> <customVmTemplate>src/test/resources/cucumber-custom-runner.vm</customVmTemplate> <!-- Specify a custom package name for generated sources. Default is no package.--> <packageName></packageName> </configuration> </execution> </executions></plugin>*Web Driver Remote Node 1 2 3 Grid HubWeb Driver Remote Node 1 2 3 Grid Hub****Single node can be used to launch multiple browser sessions - Multiple nodes can be used to obtain segregation and logging of test cases in respective browser sessions1) Register Selenium Hub****java -jar selenium-server-standalone-2.48.2.jar -port* ***5555*** *-role hub*<http://www.seleniumhq.org/download/>**2) Console Dashboard URL:** [http://localhost:5551/grid/console](http://localhost:4444/grid/console)**3) IE Node Setup**java -Dwebdriver.ie.driver=C:/eclipse/IEDriverServer/IEDriverServer.exe -jar selenium-server-standalone-2.48.2.jar -port 333**3** -role node -hub <http://localhost:5551/grid/register> -browser "browserName=internet explorer,version=11,platform=ANY,maxInstances=10"**4) Chrome Node Setup**java -Dwebdriver.chrome.driver=C:/eclipse/chromedriver/chromedriver.exe -jar selenium-server-standalone-2.48.2.jar -port **3332** -role node -hub <http://localhost:5551/grid/register> -browser "browserName=chrome, version=ANY, maxInstances=10, platform=ANY" **5) FireFox Node Setup**java -jar selenium-server-standalone-2.48.2.jar -port **3331** -role node -hub [http://localhost:5551/grid/register](http://localhost:4444/grid/register) -browser "browserName=firefox, maxInstances=10, platform=ANY, seleniumProtocol=WebDriver"java -Dwebdriver.gecko.driver="C:\geckodriver\geckodriver.exe" -jar selenium-server-standalone-3.0.1.jar -role node -hub [http://localhost:5551/grid/register](http://localhost:4444/grid/register) -browser "browserName=firefox, maxInstances=10, platform=ANY, seleniumProtocol=WebDriver"**IE, Chrome, Safari & firefox selenium NODE** java -Dwebdriver.ie.driver=C:/eclipse/IEDriverServer/IEDriverServer.exe -Dwebdriver.chrome.driver=C:/eclipse/chromedriver/chromedriver.exe -jar selenium-server-standalone-2.48.2.jar -port **5554** -role node -hub [http://localhost:5551/grid/register](http://localhost:4444/grid/register) -browser "browserName=firefox, maxInstances=10, platform=ANY, seleniumProtocol=WebDriver" -browser "browserName=internet explorer, version=11, platform=ANY, maxInstances=10" -browser "browserName=chrome,version=ANY,maxInstances=10,platform=ANY"String ipAddress = "XXX.XXX.XXX.XX";/\* Firefox \*/driver = new RemoteWebDriver(new URL("http://localhost:5551/wd/hub"), capability);/\* Chrome \*/driver = new RemoteWebDriver(new URL("http://localhost:5551/wd/hub"), capability);/\* IE \*/driver = new RemoteWebDriver(new URL("http://localhost:5551/wd/hub"), capability);/\* Code Snippet \*/DesiredCapabilities capability = DesiredCapabilities.firefox(); driver = new RemoteWebDriver(new URL("http://localhost:5551/wd/hub"), capability); driver.manage().timeouts().pageLoadTimeout(30, TimeUnit.SECONDS); driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS); Chrome Driver capabilitiesDesiredCapabilities capability = DesiredCapabilities.chrome(); driver = new RemoteWebDriver(new URL("http://localhost:5551/wd/hub"), capability);**Gecko Driver - Firefox Capabilities**System.setProperty("webdriver.gecko.driver", "C:\\Selenium\\geckodriver.exe"); DesiredCapabilities cap = DesiredCapabilities.firefox(); cap.setCapability("marionette", true); cap.setBrowserName("firefox"); driver = new RemoteWebDriver(new URL("http://192.168.117.135:5555/wd/hub"), cap);//true to enable the JS wait = new WebDriverWait(driver, 3000);**Internet Explorer Driver Capabilities**DesiredCapabilities capability =DesiredCapabilities.internetExplorer(); driver = new RemoteWebDriver(new URL("http://localhost:5551/wd/hub"), capability);