**Selenium 3 - Launching firefox browser using Geckodriver**

The first example that we will look into is launching firefox using the Geckodriver. When using Selenium 3 beta releases, you have to download geckodriver. Just like the other drivers available to Selenium, Mozilla has released geckodriver executable that will run alongside the browser.

You can download the latest executable on the [GitHub page](https://github.com/mozilla/geckodriver/releases). Now you need to specify the system property with the path

System.setProperty("webdriver.gecko.driver","path of geckodriver.exe");

WebDriver driver = **new** FirefoxDriver();

**Below is the code to set GeckoDriver path on Mac OS X for Selenium WebDriver**

System.setProperty("webdriver.gecko.driver", "/Users/username/Downloads/geckodriver");

WebDriver driver = **new** FirefoxDriver();

**What is GeckoDriver?**

A Proxy for using W3C WebDriver-compatible clients to interact with Gecko-based browsers. Geckodriver provides HTTP API described by the WebDriver protocol to communicate with Gecko browsers, such as Firefox (Version after 47).

Marionette (the next generation of FirefoxDriver) is turned on by default from Selenium 3. Even if you are working with older versions of Firefox browser, Selenium 3 expects you to set path to the driver executable by the webdriver.gecko.driver.

Click here For more details on [Marionette](https://developer.mozilla.org/en-US/docs/Mozilla/QA/Marionette/)

**Note**: If you are using Selenium version below 2.53, you don't need gecko additional driver.

If you are not doing so, it will throw exception ***"java.lang.IllegalStateException: The path to the driver executable must be set by the webdriver.gecko.driver system property;"***

java.lang.IllegalStateException: The path to the driver executable must be set by the webdriver.gecko.driver system property; for more information, see <https://github.com/mozilla/geckodriver>. The latest version can be downloaded from <https://github.com/mozilla/geckodriver/releases>

at com.google.common.base.Preconditions.checkState(Preconditions.java:199)

at org.openqa.selenium.remote.service.DriverService.findExecutable(DriverService.java:109)

at org.openqa.selenium.firefox.GeckoDriverService.access$100(GeckoDriverService.java:38)

at org.openqa.selenium.firefox.GeckoDriverService$Builder.findDefaultExecutable(GeckoDriverService.java:91)

at org.openqa.selenium.remote.service.DriverService$Builder.build(DriverService.java:296)

at org.openqa.selenium.firefox.FirefoxDriver.createCommandExecutor(FirefoxDriver.java:245)

at org.openqa.selenium.firefox.FirefoxDriver.<init>(FirefoxDriver.java:220)

at org.openqa.selenium.firefox.FirefoxDriver.<init>(FirefoxDriver.java:215)

at org.openqa.selenium.firefox.FirefoxDriver.<init>(FirefoxDriver.java:211)

at org.openqa.selenium.firefox.FirefoxDriver.<init>(FirefoxDriver.java:124)

**The other important changes in Selenium 3.x are listed below:**

\* Minimum java version is now 8+  
\* The original RC APIs are only available via the leg-rc package.  
\* To run exported IDE tests, ensure that the leg-rc package is on the classpath.  
\* Support for Firefox is via Mozilla's geckodriver.  
\* Support for Edge is provided by MS:  
\* Official support for IE requires version 9 or above  
\* New html-table runner backed by WebDriver.  
\* Unused command line arguments are now no longer parsed.

Now let us see the example to launch firefox browser with Selenium 3 using gecko driver.

**package** com.test;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.firefox.FirefoxDriver;

**import** org.testng.annotations.Test;

**public** **class** **ExampleThree** {

String driverPath = "<path to gecko driver executable>";

**public** WebDriver driver;

@Test

**public** **void** **launchBrowser**() {

System.out.println("launching firefox browser");

System.setProperty("webdriver.gecko.driver", driverPath+"geckodriver.exe");

driver = **new** FirefoxDriver();

}

@Test

**public** **void** **openApplication**() {

driver.navigate().to("[http://www.google.com"](http://www.google.com/));

}

@Test

**public** **void** **closeDriver**() {

**if**(driver!=**null**) {

driver.close();

}

}

}

To run tests on remote machines, WebDriver has to use the instance of the RemoteWebDriver and DesiredCapabilities in order to specify browser name, version and platform to execute tests.

Generally to run tests on our local machine, we will just specify as WebDriver driver = new FirefoxDriver(); to run on Firefox browser.

To execute tests on remote machine, we need to use remotewebdriver, below is the sample code to execute your tests on remote machine with Firefox gecko driver

System.setProperty("webdriver.gecko.driver", driverPath+"geckodriver.exe");

DesiredCapabilities capabilities=DesiredCapabilities.firefox();

capabilities.setCapability("marionette", **true**);

WebDriver driver = **new** FirefoxDriver(capabilities);

The above code is verified with selenium-server-standalone-3.0.0-beta2 and Firefox 48 version. Please let me know if you have any issues.

In your Maven project, just add / update the following selenium dependency to your pom.xml:

    <**dependency**>

        <**groupId**>org.seleniumhq.selenium</**groupId**>

        <**artifactId**>selenium-java</**artifactId**>

        <**version**>3.0.0-beta2</**version**>

    </**dependency**>

The most common issue people are facing with latest versions of Firefox is org.openqa.selenium.firefox.NotConnectedException: Unable to connect to host 127.0.0.1 on port 7055 after 45000 ms.

Users who are facing the above problem, Please use Marionette (geckodriver).

Please do comment your issue / observation with the versions (Selenium, Firefox and Geckodriver) that you have used.

**UPDATE on geckodriver windows 32bit issues**

Sep 1, 2016 3:07:19 PM org.openqa.selenium.remote.ProtocolHandshake createSession

INFO: Attempting bi-dialect session, assuming Postel's Law holds true on the remote **end**

If you are facing an issue like **org.openqa.selenium.remote.ProtocolHandshake createSession** with geckodriver windows 32bit , Please check for an update of the issue here [and check the last comment made by 'Andreas Tolfsen'. Once all the issues fixed in the](https://github.com/mozilla/geckodriver/pull/213) [milestone](https://github.com/mozilla/geckodriver/milestone/1) they will soon be releasing Geckodriver v0.11.

**UPDATE:**  
Now you should be able to download [gecko driver version 0.11](https://github.com/mozilla/geckodriver/releases) which will resolve windows32 bit issues.

* [‹ JMeter HTTP Request Sampler Example](http://www.seleniumeasy.com/jmeter-tutorials/http-request-sampler-example)
* [Selenium 3 - Launch Edge browser using Microsoft Edge Driver ›](http://www.seleniumeasy.com/selenium-tutorials/launching-microsoft-edge-browser-using-selenium-3)

Tags:

[Selenium 3](http://www.seleniumeasy.com/tags/selenium-3)