Selenium – First Test Case Using GeckoDriver

Lecture Notes

A Complete Guide for Beginners

Softech Solutions Inc.

www.softechnosolutionsgroup.com

saney.alam@softechnosolutionsgroup.com

Selenium – First Test Case Using GeckoDriver

Table of Contents	*Page*
What is GeckoDriver	*02*
How to use GeckoDriver to launch Firefox?	*04*
Exceptions in using GeckoDriver	*07*

Mozilla runs on Gecko browser engine. Gecko browser engine was developed by Mozilla foundation as a part of Mozilla browser. However, Gecko browser engine is not limited to Mozilla browser. It is an open source browser engine which can be used by anyone in their application. It can help applications render web pages. Just like other browsers, Chrome, Internet explorer and Edge Mozilla foundation has decided to introduce Marionette driver to control Firefox browser.

Moving forward, it is expected that to interact with Firefox Browser we will need to run an instance of Marionette driver. Marionette driver, sometimes loosely termed as Gecko driver, can drive both UI and Web Page in the Firefox browser. This gives an exceptional amount of control over the Web Page and the UI of the web browser to the tester.

Selenium 3 has upgraded itself to now launch Firefox driver using Marionette driver instead of the default initialization supported earlier.

Main questions that will be answered in this tutorial are

- ***How to use Firefox Gecko Driver in Selenium***
- ***How to use Firefox Marionette Driver in Selenium***.

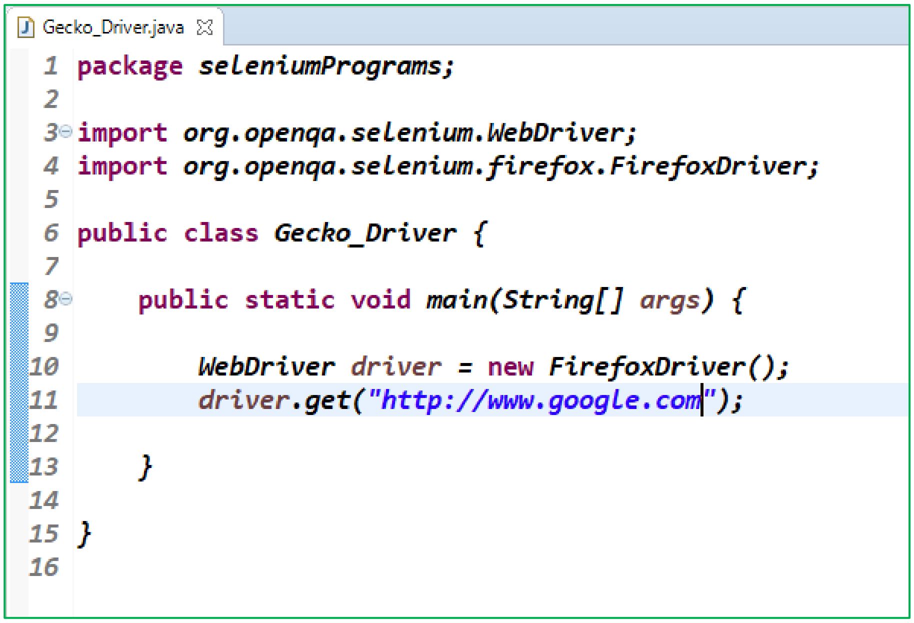
What is GeckoDriver?

Let us first start with the very basics – What is Gecko and GeckoDriver? Gecko is a web browser engine used in many applications developed by Mozilla Foundation and the Mozilla Corporation.

Gecko Driver is the link between your tests in Selenium and the Firefox browser. GeckoDriver is a proxy for using W3C WebDriver-compatible clients to interact with Gecko-based browsers i.e. Mozilla Firefox in this case. As ***Selenium 3*** will not have any native implementation of Firefox, we have to direct all the driver commands through Gecko Driver. Gecko Driver is an executable file that we need to have in one of the system path before starting our tests. Firefox browser implements the ***WebDriver*** protocol using an executable called **GeckoDriver.exe**. This executable starts a server on our system. All our tests communicate to this server to run our tests. It translates calls into the Marionette automation protocol by acting as a proxy between the local and remote ends. It is something similar for Internet explorer **&** Chrome.

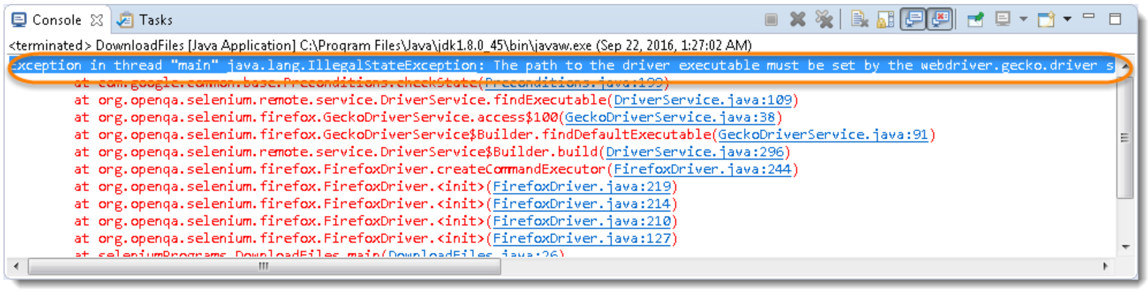
Problem Statement: Selenium 3 Test without GeckoDriver.exe

Let's just see first that what happens, if we run the Selenium Test without GeckoDriver with Selenium 3. Actually if we are new to Selenium and we have started directly with Selenium 3.0, we would not know how Firefox was launched with the previous versions of Selenium (version 2.53 and before). It was a pretty straight forward process where we were not required to use GeckoDriver or any other driver. We just write the code to instantiate the WebDriver and open Firefox. Below is the simple program to launch a Firefox browser with Selenium 2.



```java
package seleniumPrograms;

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

public class Gecko_Driver {

    public static void main(String[] args) {

        WebDriver driver = new FirefoxDriver();
        driver.get("http://www.google.com");

    }

}
```

Note: If we run the above program with ***Selenium 3***, it will throw the ***java.lang.IllegalStateException.*** It clearly says that ‘***The path to the driver executable must be set by webdriver.gecko.driver system propeerty:*** ‘.



```
<terminated> DownloadFiles [Java Application] C:\Program Files\Java\jdk1.8.0_45\bin\javaw.exe (Sep 22, 2016, 1:27:02 AM)
xception in thread "main" java.lang.IllegalStateException: The path to the driver executable must be set by the webdriver.gecko.driver s
        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:244)
        at org.openqa.selenium.firefox.FirefoxDriver.<init>(FirefoxDriver.java:219)
        at org.openqa.selenium.firefox.FirefoxDriver.<init>(FirefoxDriver.java:214)
        at org.openqa.selenium.firefox.FirefoxDriver.<init>(FirefoxDriver.java:210)
        at org.openqa.selenium.firefox.FirefoxDriver.<init>(FirefoxDriver.java:127)
        at seleniumPrograms.DownloadFiles.main(DownloadFiles.java:26)
```

Solution: Use GeckoDriver with Selenium 3

It is easy to switch to the new MarionetteDriver. I suggest you try it on your code and if your test works then keep using it. If your test fails then it is most likely some defects in the newer MarionetteDriver. In that case, it is better to raise defects to the Mozilla team and for the time being switch back to Firefox 45 or 46.

How to use GeckoDriver to launch Firefox?

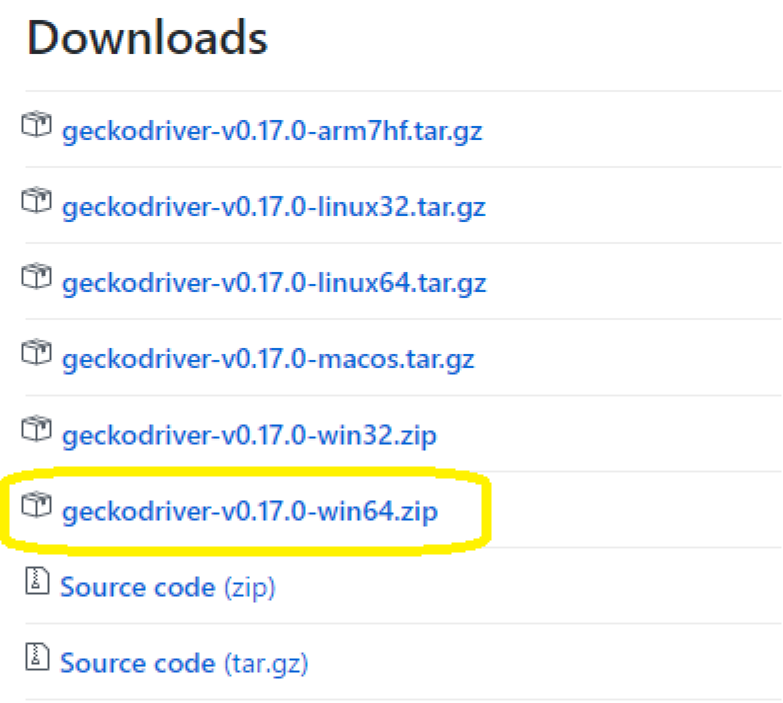
We have now come to the part where you will see how you can use GeckoDriver to launch Firefox. You will first need to download GeckoDriver and then set its path. There are three different ways to use GeckoDriver with Selenium 3:

1. **With setting system properties in the test**
2. **With setting system properties by Environment Variable**
3. **With setting up Browser Desired Capabilities**

Download Gecko Driver:-

Step 1-

Gecko Driver different versions can be downloaded from "https://github.com/mozilla/geckodriver/releases"**.** I suggest you to use the latest version.



Downloads

- geckodriver-v0.17.0-arm7hf.tar.gz
- geckodriver-v0.17.0-linux32.tar.gz
- geckodriver-v0.17.0-linux64.tar.gz
- geckodriver-v0.17.0-macos.tar.gz
- geckodriver-v0.17.0-win32.zip
- geckodriver-v0.17.0-win64.zip
- Source code (zip)
- Source code (tar.gz)

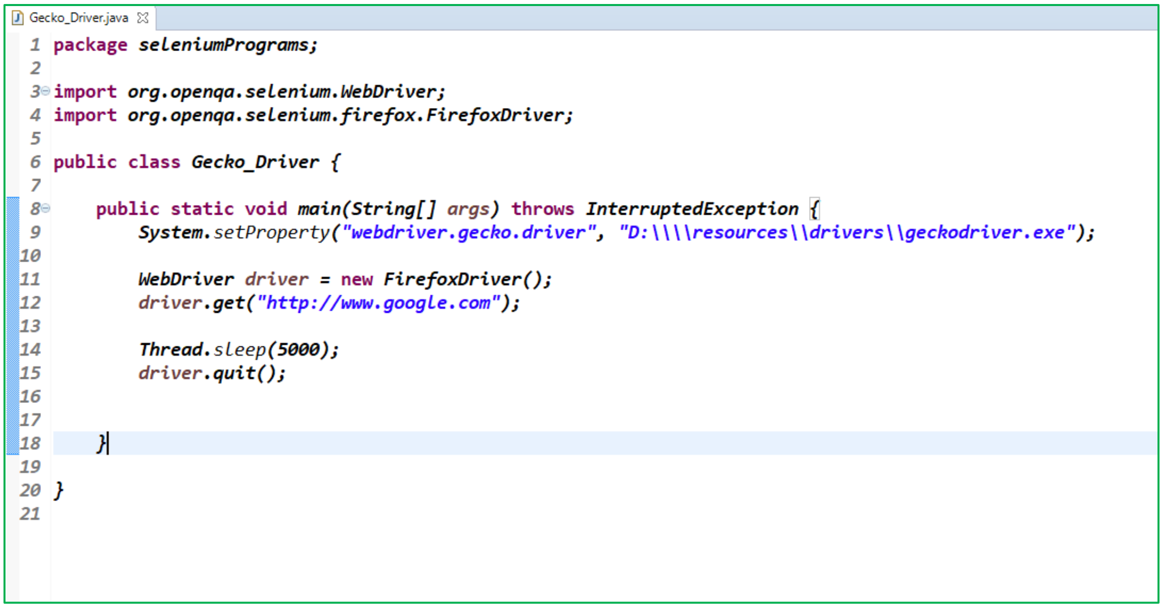
Step 2-

Once the download complete extract the downloaded folder.

This completes the downloading process. Now let's see how we can use it in our project. There are 2 methods using which we can configure this driver in our project. We can use any of these methods.

Method 1: Set System Properties for Gecko Driver:-

Code to set the System properties is
System.setProperty("webdriver.gecko.driver","Path to geckodriver.exe");
The complete program to launch the GeckoDriver will be like this:



```java
package seleniumPrograms;

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

public class Gecko_Driver {

    public static void main(String[] args) throws InterruptedException {
        System.setProperty("webdriver.gecko.driver", "D:\\\\resources\\drivers\\geckodriver.exe");

        WebDriver driver = new FirefoxDriver();
        driver.get("http://www.google.com");

        Thread.sleep(5000);
        driver.quit();


    }

}

```

__Note-__ No needs to set the system properties if you are still using Selenium 2.
Run this code to verify that everything is working fine. You will notice that google.com is opened in new Firefox window.

Method 2: Set property in Environment Variables:-

Step 1:

Go to My Computer and Right Click to get the context menu and then select Properties.

Step 2:

Click on the **Change Settings** on the opened window.

Step 3:

Go to **Advance tab** in the System Properties window and click on **Environment Variables**.

Step 4:

Now under the System variables, select **Path** and click on **Edit**.

Step 5:

Click New and paste the path of the GeckoDriver and Click Ok.

Note: Once the path is set, you would not need to set the System property every time in the test script. Your test script would simply work without the System Property code.

Set Desired Capabilities of the Marionette/Gecko Driver

To use Marionette in our tests we will need to update our desired capabilities to use it.

```
package seleniumPrograms;

        import org.openqa.selenium.WebDriver;
        import org.openqa.selenium.firefox.FirefoxDriver;
        import org.openqa.selenium.remote.DesiredCapabilities;
public class Gecko_Driver {

        public static void main(String[] args) throws InterruptedException {
                DesiredCapabilities capabilities =
DesiredCapabilities.firefox();
                capabilities.setCapability("marionette", true);
                WebDriver driver = new FirefoxDriver(capabilities);
                driver.get("http://www.google.com");

                Thread.sleep(5000);
                driver.quit();
        }
}
```

Exceptions in using GeckoDriver

There multiple exceptions you can face by using the GeckoDriver:

Error 1:

***Exception**: Exception in thread "main" org.openqa.selenium.WebDriverException: Failed to decode response from marionette (WARNING: The server did not provide any stacktrace information).*

***Solution**: There is some mismatch between the Gecko version or Firefox version or Selenium version. First try updating the Gecko version from **Github**. If you still face the error try updating the FF version as well.*

Error 2:

***Exception:** Exception in thread "main" org.openqa.selenium.remote.UnreachableBrowserException: Error communicating with the remote browser. It may have died.*

***Solution**: This happens to me because I was using driver.quit(). So avoid using this and replace this statement with driver.close().*