**Parallel Testing without Selenium Grid (On your Local PC)**

<="" ins="">

In this article, I will describe how to do **parallel testing on your PC (local computer) without using Selenium Grid**. There are several techniques to do this operation and I will share with you some code samples.

In my former articles, I described parallel testing theory so in this tutorial I do not want to go into details. I will give the test scenarios and their implementations. Also, I will describe the details of each test files and I will add informative comments in my test automation codes.

**Test Scenario for Parallel Testing**

* We will have two test classes.
* First Class has three methods and all methods must open Google and check its title. First class’s test methods must run on Firefox.
* Second Class has two methods and first test method must open Google and checks its title, second test method opens Yandex and checks its title. Second class’s test methods must run on Chrome.
* All tests have to Run in Parallel in Locally!

**Test Design:**

**1) OptionsManager Class:**

This class has a “**getChromeOptions**” and “**getFifefoxOptions**” methods and they take browser name as a parameter then they provide related options as per specific browser types.

OptionsManager

Java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37 | package com;    import org.openqa.selenium.chrome.ChromeOptions;  import org.openqa.selenium.firefox.FirefoxDriver;  import org.openqa.selenium.firefox.FirefoxOptions;  import org.openqa.selenium.firefox.FirefoxProfile;    public class OptionsManager {        //Get Chrome Options      public ChromeOptions getChromeOptions() {          ChromeOptions options = new ChromeOptions();          options.addArguments("--start-maximized");          options.addArguments("--ignore-certificate-errors");          options.addArguments("--disable-popup-blocking");          //options.addArguments("--incognito");          return options;          /\*ChromeDriverService service = new ChromeDriverService.Builder()                  .usingAnyFreePort()                  .build();          ChromeDriver driver = new ChromeDriver(service, options);\*/      }        //Get Firefox Options      public FirefoxOptions getFirefoxOptions () {          FirefoxOptions options = new FirefoxOptions();          FirefoxProfile profile = new FirefoxProfile();          //Accept Untrusted Certificates          profile.setAcceptUntrustedCertificates(true);          profile.setAssumeUntrustedCertificateIssuer(false);          //Use No Proxy Settings          profile.setPreference("network.proxy.type", 0);          //Set Firefox profile to capabilities          options.setCapability(FirefoxDriver.PROFILE, profile);          return options;      }  } |

**2) ThreadLocal Driver Factory Class:**

First of all, for Thread-Safe test automation, I will use **ThreadLocal<WebDriver>** and I will create a ThreadLocalDriverFactory (TLDriverFactory) class to **set ThreadLocal driver** and get **Webdriver**.

TLDriverFactory.java

Java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28 | package com;    import org.openqa.selenium.WebDriver;  import org.openqa.selenium.chrome.ChromeDriver;  import org.openqa.selenium.firefox.FirefoxDriver;  import org.openqa.selenium.remote.DesiredCapabilities;    /\*\*  \* Created by onurb on 19-Dec-16.  \* Updated by onurb on 10-Nov-17.  \*/  public class TLDriverFactory {        OptionsManager optionsManager = new OptionsManager();      private static ThreadLocal<WebDriver> tlDriver = new ThreadLocal<>();        public synchronized void setTLDriver (String browser) {          if (browser.equals("firefox")) {              tlDriver = ThreadLocal.withInitial(() -> new FirefoxDriver(optionsManager.getFirefoxOptions()));          } else if (browser.equals("chrome")) {              tlDriver = ThreadLocal.withInitial(() -> new ChromeDriver(optionsManager.getChromeOptions()));          }      }        public synchronized ThreadLocal<WebDriver> getTLDriver () {          return tlDriver;      }  } |

**3) TestBase Class:**

In Base Test class, I  instantiated *Thread Local Driver Factory* class and **@BeforeMethod** I got **browser** parameter **from TestNG.xml** file (I will describe it soon),  and I will set and get the driver by using ThreadLocalMap which is declared in ThreadLocalDriverFactory class. **@AfterMethod**, I closed the driver.

TestBase

Java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34 | package com;    import org.openqa.selenium.WebDriver;  import org.openqa.selenium.remote.DesiredCapabilities;  import org.openqa.selenium.support.ui.WebDriverWait;  import org.testng.annotations.\*;    import java.net.MalformedURLException;    /\*\*  \* Created by ONUR on 03.12.2016.  \*/  public class TestBase {        protected WebDriver driver;      protected WebDriverWait wait;      private TLDriverFactory TLDriverFactory = new TLDriverFactory();        //Do the test setup      @BeforeMethod      @Parameters(value={"browser"})      public void setupTest (@Optional String browser) throws MalformedURLException {          //Set & Get ThreadLocal Driver with Browser          TLDriverFactory.setTLDriver(browser);          driver = TLDriverFactory.getTLDriver().get();          wait = new WebDriverWait(driver, 15);      }        @AfterMethod      public synchronized void tearDown() throws Exception {          driver.quit();          TLDriverFactory.getTLDriver().remove();      }  } |

**4) FirstTest Class:**

FirstTest

Java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46 | package com;    import org.testng.Assert;  import org.testng.annotations.Test;    /\*\*  \* Created by ONUR on 03.12.2016.  \*/  public class FirstTest extends TestBase {        @Test      public void GOOGLE0() throws Exception {          System.out.println("Google0 Test Started! " + Thread.currentThread().getId());          driver.navigate().to("http://www.google.com");          System.out.println("Google0 Test's Page title is: " + driver.getTitle() + " " + Thread.currentThread().getId());          Assert.assertEquals(driver.getTitle(), "Google");          System.out.println("Google0 Test Ended! " + Thread.currentThread().getId());      }        @Test      public void GOOGLE2() throws Exception {          System.out.println("Google2 Test Started! " + Thread.currentThread().getId());          driver.navigate().to("http://www.google.com");          System.out.println("Google2 Test's Page title is: " + driver.getTitle() + " " + Thread.currentThread().getId());          Assert.assertEquals(driver.getTitle(), "Google");          System.out.println("Google2 Test Ended! " + Thread.currentThread().getId());      }        @Test      public void GOOGLE3() throws Exception {          System.out.println("Google3 Test Started! " + Thread.currentThread().getId());          driver.navigate().to("http://www.google.com");          System.out.println("Google3 Test's Page title is: " + driver.getTitle() + " " + Thread.currentThread().getId());          Assert.assertEquals(driver.getTitle(), "Google");          System.out.println("Google3 Test Ended! " + Thread.currentThread().getId());      }        @Test      public void GOOGLE4() throws Exception {          System.out.println("Google4 Test Started! " + Thread.currentThread().getId());          driver.navigate().to("http://www.google.com");          System.out.println("Google3 Test's Page title is: " + driver.getTitle() + " " + Thread.currentThread().getId());          Assert.assertEquals(driver.getTitle(), "Google");          System.out.println("Google3 Test Ended! " + Thread.currentThread().getId());      }  } |

**5) SecondTest Class:**

SecondTest

Java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28 | package com;    import org.testng.Assert;  import org.testng.annotations.Test;    /\*\*  \* Created by ONUR on 03.12.2016.  \*/  public class SecondTest extends TestBase{        @Test      public void GOOGLE1() throws Exception {          System.out.println("Google1 Test Started! " + Thread.currentThread().getId());          driver.navigate().to("http://www.google.com");          System.out.println("Google1 Test's Page title is: " + driver.getTitle() + " " + Thread.currentThread().getId());          Assert.assertEquals(driver.getTitle(), "Google");          System.out.println("Google1 Test Ended! " + Thread.currentThread().getId());      }        @Test      public void YANDEX() throws Exception {          System.out.println("Yandex Test Started! " + Thread.currentThread().getId());          driver.navigate().to("http://www.yandex.com");          System.out.println("Yandex Test's Page title is: " + driver.getTitle() + " " + Thread.currentThread().getId());          Assert.assertEquals(driver.getTitle(), "Yandex");          System.out.println("Yandex Test Ended! " + Thread.currentThread().getId());      }  } |

**6) TestNG.xml File:**

TestNG.xml

XHTML

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25 | <?xml version="1.0" encoding="UTF-8"?>  <!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">  <suite thread-count="2" name="Suite" parallel="tests" >      <test name="com.FirstTest" parallel="methods" thread-count="5">          <parameter name="browser" value="firefox"/>          <classes>              <class name="com.FirstTest">                  <methods>                      <include name="GOOGLE0" />                      <include name="GOOGLE2" />                  </methods>              </class>          </classes>      </test> <!-- First Test -->      <test name="com.SecondTest"  parallel="methods" thread-count="4">          <parameter name="browser" value="chrome"/>          <classes>              <class name="com.SecondTest">                  <methods>                      <include name="GOOGLE1" />                  </methods>              </class>          </classes>      </test> <!-- Second Test -->  </suite> <!-- Suite --> |

**7) POM.xml:**

pom.xml

XHTML

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37 | <?xml version="1.0" encoding="UTF-8"?>  <project xmlns="http://maven.apache.org/POM/4.0.0"           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"           xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">      <modelVersion>4.0.0</modelVersion>        <groupId>TestNGParallel</groupId>      <artifactId>TestNGParallel</artifactId>      <version>1.0-SNAPSHOT</version>      <build>          <plugins>              <plugin>                  <groupId>org.apache.maven.plugins</groupId>                  <artifactId>maven-compiler-plugin</artifactId>                  <configuration>                      <source>1.8</source>                      <target>1.8</target>                  </configuration>              </plugin>          </plugins>      </build>        <dependencies>          <dependency>              <groupId>org.seleniumhq.selenium</groupId>              <artifactId>selenium-java</artifactId>              <version>RELEASE</version>              <scope>test</scope>          </dependency>            <dependency>              <groupId>org.testng</groupId>              <artifactId>testng</artifactId>              <version>RELEASE</version>          </dependency>      </dependencies>  </project> |

[**Click here to go to Github Link**](https://github.com/swtestacademy/TestNGParallel/tree/NewParallelTestNGImplementation)

**LATEST NOTES**

If you have several test classes and in those test classes if you have more than one test methods, then you can run those test classes and methods in parallel with below TestNG configuration.

XHTML

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27 | <?xml version="1.0" encoding="UTF-8"?>  <!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">  <suite thread-count="2" name="Suite" parallel="tests" >      <test name="com.FirstTest" parallel="methods" thread-count="5">          <parameter name="browser" value="chrome"/>          <parameter name="platform" value="WINDOWS"/>          <classes>              <class name="com.FirstTest">                  <methods>                      <include name="GOOGLE0" />                      <include name="GOOGLE2" />                  </methods>              </class>          </classes>      </test> <!-- First Test -->      <test name="com.SecondTest"  parallel="methods" thread-count="4">          <parameter name="browser" value="chrome"/>          <parameter name="platform" value="WINDOWS"/>          <classes>              <class name="com.SecondTest">                  <methods>                      <include name="GOOGLE1" />                  </methods>              </class>          </classes>      </test> <!-- Second Test -->  </suite> <!-- Suite --> |

Above configuration runs two parallel test classes because of below line.

**<suite thread-count=”2″ name=”Suite” parallel=”tests” >**

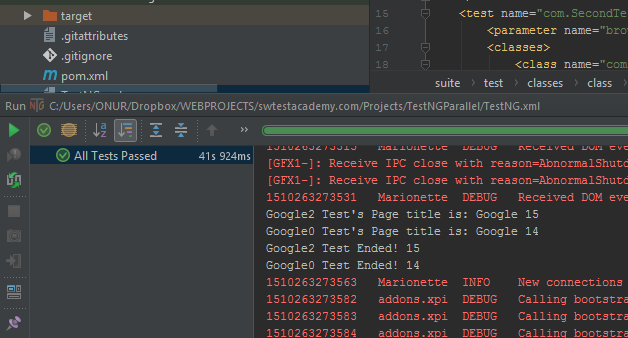
And it also runs five test methods of test class one in parallel too by using below line.

**<test name=”com.FirstTest” parallel=”methods” thread-count=”5″>**

And it also runs four test methods of test class two in parallel too by using below line.

**<test name=”com.SecondTest” parallel=”methods” thread-count=”4″>**

**Here is the result for parallel testing in local machine!**



**Execution Order of Parallel Testing in Local Computer:**

Google2 Test Started! 15  
Google1 Test Started! 13  
Google0 Test Started! 14  
Google1 Test’s Page title is: Google 13  
Google1 Test Ended! 13  
Google2 Test’s Page title is: Google 15  
Google0 Test’s Page title is: Google 14  
Google2 Test Ended! 15  
Google0 Test Ended! 14

**Selenium TestNG Parallel Execution UPDATES! [03.04.2018]**

After ChromeDriver 2.36, above implementation has some problems (ChromeDriver 2.35 is working with above implementation). The driver object in BaseTest class is overwritten by the latest value of ThreadLocal driver’s getDriver() method. Thus, we need to create specific Webdriver and Wait objects for each class. I tried several solutions but the following implementation worked flawlessly so I will share the codes below. If you have some problems with the above implementation please try the below one. I also added InvokedMethodListener for before and after method implementation. Please, go at the end of the following post. I don’t want to create a duplicate content.