**Test Scenario:**

We have two test classes. The first one has three test methods which open www.google.com and check that title is “Google”. The second one has two test methods and they open Google and Yandex and then check their titles. Tests are very simple but our aim is to run them in parallel with different browsers. We will run the first test with Chrome and the second one with Firefox. We will configure this within TestNG.xml file.

**Making Your TestNG Tests Thread-Safe:**

It is very important to construct our tests **thread-safe in order to run them in parallel without a problem**. We have to make sure that **shared resources are isolated within each thread**. Thus, we need to initialize all related resources within test method. Also, we need to keep test specific resources thread local and keep your static class members as static **that is really need to be static**. Apply these to all the classes that are loading during the test execution.

**Test Architecture:**

I used three JAVA files for our test. These are **FirstTest.java**, **SecondTest.java** and **TestBase.java**. Also, I did the configurations in **TestNG.xml** file.

**TestBase.java**, I created **ThreadLocal <>() webdriver (ThreadLocalMap)** for thread-safe test execution and I got **TestNG parameter (browser)** with **@Parameter** annotation. In setupTest method, I created and configured Desired Capabilities and set our local grid address. **getDriver()** method returns the created driver. **FirstTest** and **SecondTest** classes extend TestBase class and comprise of their test code.

**Test Code:**

**Before run the test, you need to trigger Selenium Grid!  
It is described at first section of** [**this article**](https://www.swtestacademy.com/parallel-tests-selenium-grid-junit/)**.**

**TestBase.java**

TestBase.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  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58 | import org.openqa.selenium.WebDriver;  import org.openqa.selenium.firefox.FirefoxDriver;  import org.openqa.selenium.firefox.FirefoxProfile;  import org.openqa.selenium.remote.DesiredCapabilities;  import org.openqa.selenium.remote.RemoteWebDriver;  import org.testng.annotations.AfterClass;  import org.testng.annotations.AfterMethod;  import org.testng.annotations.BeforeMethod;  import org.testng.annotations.Parameters;    import java.net.MalformedURLException;  import java.net.URL;    public class TestBase {        //Declare ThreadLocal Driver (ThreadLocalMap) for ThreadSafe Tests      protected static ThreadLocal<RemoteWebDriver> driver = new ThreadLocal<>();        @BeforeMethod      @Parameters(value={"browser"})      public void setupTest (String browser) throws MalformedURLException {          //Set DesiredCapabilities          DesiredCapabilities capabilities = new DesiredCapabilities();            //Firefox Profile Settings          /\*if (browser=="firefox") {              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              capabilities.setCapability(FirefoxDriver.PROFILE, profile);          }\*/            //Set BrowserName          capabilities.setCapability("browserName", browser);            //Set Browser to ThreadLocalMap          driver.set(new RemoteWebDriver(new URL("http://localhost:4444/wd/hub"), capabilities));      }        public WebDriver getDriver() {          //Get driver from ThreadLocalMap          return driver.get();      }        @AfterMethod      public void tearDown() throws Exception {          getDriver().quit();      }        @AfterClass void terminate () {          //Remove the ThreadLocalMap element          driver.remove();      }  } |

**FirstTest.java**

FirstTest.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  29  30  31  32  33  34  35 | import org.testng.Assert;  import org.testng.annotations.Test;    /\*\*  \* Created by ONUR on 03.12.2016.  \*/  public class FirstTest extends TestBase {        @Test      public void GOOGLE1() throws Exception {          System.out.println("Google1 Test Started! " + Thread.currentThread().getId());          getDriver().navigate().to("http://www.google.com");          System.out.println("Google1 Test's Page title is: " + getDriver().getTitle() +" "+ Thread.currentThread().getId());          Assert.assertEquals(getDriver().getTitle(), "Google");          System.out.println("Google1 Test Ended! " + Thread.currentThread().getId());      }        @Test      public void GOOGLE2() throws Exception {          System.out.println("Google2 Test Started! " + Thread.currentThread().getId());          getDriver().navigate().to("http://www.google.com");          System.out.println("Google2 Test's Page title is: " + getDriver().getTitle() +" "+ Thread.currentThread().getId());          Assert.assertEquals(getDriver().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());          getDriver().navigate().to("http://www.google.com");          System.out.println("Google3 Test's Page title is: " + getDriver().getTitle() +" "+ Thread.currentThread().getId());          Assert.assertEquals(getDriver().getTitle(), "Google");          System.out.println("Google3 Test Ended! " + Thread.currentThread().getId());      }  } |

**SecondTest.java**

SecondTest.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 | import org.testng.Assert;  import org.testng.annotations.Test;    /\*\*  \* Created by ONUR on 03.12.2016.  \*/  public class SecondTest extends TestBase{        @Test      public void GOOGLE4() throws Exception {          System.out.println("Google4 Test Started! " + Thread.currentThread().getId());          getDriver().navigate().to("http://www.google.com");          System.out.println("Google4 Test's Page title is: " + getDriver().getTitle() +" "+ Thread.currentThread().getId());          Assert.assertEquals(getDriver().getTitle(), "Google");          System.out.println("Google4 Test Ended! " + Thread.currentThread().getId());      }        @Test      public void YANDEX() throws Exception {          System.out.println("Yandex Test Started! " + Thread.currentThread().getId());          getDriver().navigate().to("http://www.yandex.com");          System.out.println("Yandex Test's Page title is: " + getDriver().getTitle() +" "+ Thread.currentThread().getId());          Assert.assertEquals(getDriver().getTitle(), "Yandex");          System.out.println("Yandex Test Ended! " + Thread.currentThread().getId());      }  } |

**TestNG.xml**

TestNG.xml

XHTML

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

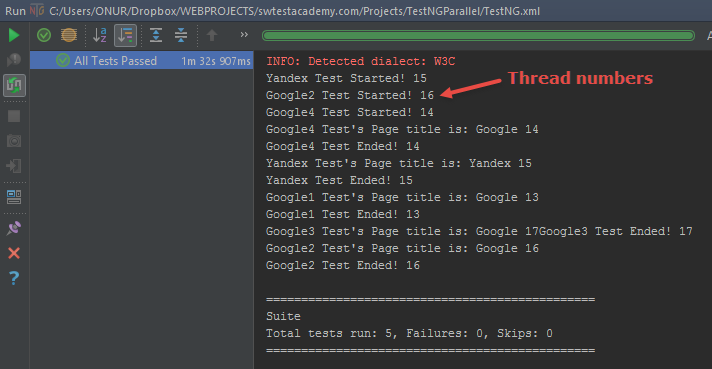
**pom.xml**

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 | <?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>        <dependencies>          <dependency>              <groupId>org.seleniumhq.selenium</groupId>              <artifactId>selenium-java</artifactId>              <version>RELEASE</version>          </dependency>            <dependency>              <groupId>org.testng</groupId>              <artifactId>testng</artifactId>              <version>RELEASE</version>          </dependency>      </dependencies>    </project> |

**Test Results:**

[](https://www.swtestacademy.com/wp-content/uploads/2017/11/img_5a048886a1a82.png)

***Github:*** [***https://github.com/swtestacademy/TestNGParallel***](https://github.com/swtestacademy/TestNGParallel)

**Selenium TestNG Parallel Execution Latest 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.

**TLDriverFactory**

TLDriverFactory

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  47  48  49  50 | 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;  import org.openqa.selenium.remote.RemoteWebDriver;  import org.openqa.selenium.support.ui.WebDriverWait;    import java.net.MalformedURLException;  import java.net.URL;      public class TLDriverFactory {        private static OptionsManager optionsManager = new OptionsManager();      private static ThreadLocal<WebDriver> tlDriver = new ThreadLocal<>();        public static synchronized void setDriver(String browser) {          if (browser.equals("firefox")) {              //For Local Usage              tlDriver = ThreadLocal.withInitial(() -> new FirefoxDriver(optionsManager.getFirefoxOptions()));                //For Grid Usage              /\*try {                  tlDriver.set(new RemoteWebDriver(new URL("http://localhost:4444/wd/hub"), optionsManager.getFirefoxOptions()));              } catch (MalformedURLException e) {                  e.printStackTrace();              }\*/          } else if (browser.equals("chrome")) {              //For Local Usage              tlDriver.set(new ChromeDriver(optionsManager.getChromeOptions()));                /\*//For Grid Usage              try {                  tlDriver.set(new RemoteWebDriver(new URL("http://localhost:4444/wd/hub"), optionsManager.getChromeOptions()));              } catch (MalformedURLException e) {                  e.printStackTrace();              }\*/          }      }        public static synchronized WebDriverWait getWait (WebDriver driver) {          return new WebDriverWait(driver,20);      }        public static synchronized WebDriver getDriver() {          return tlDriver.get();      }  } |

**TestBase (I commented out everything. InvokedMethodListener does before/after method operations.)**

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 | package com;    import org.openqa.selenium.WebDriver;  import org.openqa.selenium.chrome.ChromeDriver;  import org.openqa.selenium.support.ui.WebDriverWait;  import org.testng.annotations.\*;    import javax.swing.text.html.Option;  import java.net.MalformedURLException;  import java.util.Map;  import java.util.concurrent.ConcurrentHashMap;    /\*\*  \* Created by ONUR on 03.12.2016.  \*/  public class TestBase {        //public WebDriver driver;      //public Map<Long, WebDriver> driverMap = new ConcurrentHashMap();      //public WebDriverWait wait;      //public TLDriverFactory tlDriverFactory = new TLDriverFactory();        //Do the test setup      @BeforeMethod      @Parameters(value={"browser"})      public void setupTest (@Optional String browser) throws MalformedURLException {          //System.out.println("BeforeMethod is started. " + Thread.currentThread().getId());          //Set & Get ThreadLocal Driver with Browser          //TLDriverFactory.setDriver(browser);          //driver = TLDriverFactory.getDriver();          //driverMap.put(Thread.currentThread().getId(),TLDriverFactory.getDriver());          //driver = driverMap.get(Long.valueOf(Thread.currentThread().getId()));          //wait = new WebDriverWait(driver, 15);      }        @AfterMethod      public void tearDown() throws Exception {          //System.out.println("AfterMethod is started. " + Thread.currentThread().getId());          //TLDriverFactory.getDriver().quit();          //TLDriverFactory.getTLDriver().remove();      }    } |

**InvokedMethodListener (This does before and after test method operations.)**

InvokedMethodListener

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 | import org.testng.IInvokedMethodListener;  import org.testng.ITestResult;    public class InvokedMethodListener implements IInvokedMethodListener {      @Override      public void beforeInvocation(IInvokedMethod method, ITestResult testResult) {          if (method.isTestMethod()) {              System.out.println("Test Method BeforeInvocation is started. " + Thread.currentThread().getId());              String browserName = method.getTestMethod().getXmlTest().getLocalParameters().get("browser");              TLDriverFactory.setDriver(browserName);          }      }        @Override      public void afterInvocation(IInvokedMethod method, ITestResult testResult) {          if (method.isTestMethod()) {              System.out.println("Test Method AfterInvocation is started. " + Thread.currentThread().getId());              WebDriver driver = TLDriverFactory.getDriver();              if (driver != null) {                  driver.quit();              }          }      }  } |

**OptionsManager**

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;      }  } |

**FirstTest**

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  47  48  49  50  51  52  53  54 | package com;    import org.openqa.selenium.WebDriver;  import org.openqa.selenium.support.ui.WebDriverWait;  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 {          WebDriver driver = TLDriverFactory.getDriver();          WebDriverWait wait = TLDriverFactory.getWait(driver);          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 {          WebDriver driver = TLDriverFactory.getDriver();          WebDriverWait wait = TLDriverFactory.getWait(driver);          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 {          WebDriver driver = TLDriverFactory.getDriver();          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 {          WebDriver driver = TLDriverFactory.getDriver();          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());      }  } |

**SecondTest**

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  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43 | package com;    import org.openqa.selenium.WebDriver;  import org.testng.Assert;  import org.testng.annotations.Test;    /\*\*  \* Created by ONUR on 03.12.2016.  \*/  public class SecondTest extends TestBase{        WebDriver driver;        private WebDriver getDriver () {          return driver = TLDriverFactory.getDriver();      }        @Test      public void GOOGLE1() throws Exception {          WebDriver driver = TLDriverFactory.getDriver();          System.out.println("Google1 Test Started! " + Thread.currentThread().getId());          driver.navigate().to("http://www.google.com");          //System.out.println("GOOGLE1-TEST - TLDriverFactory.getDriver(): " + driver);          System.out.println("GOOGLE1-TEST - driver: " + driver);          //System.out.println("Map Driver Google: " + driverMap.get(Long.valueOf(Thread.currentThread().getId())));          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 {          WebDriver driver = TLDriverFactory.getDriver();          System.out.println("Yandex Test Started! " + Thread.currentThread().getId());          driver.navigate().to("http://www.yandex.com");          //System.out.println("YANDEX-TEST - TLDriverFactory.getDriver(): " + driver);          System.out.println("YANDEX-TEST - driver: " + driver);          //System.out.println("Map Driver Yandex: " + driverMap.get(Long.valueOf(Thread.currentThread().getId())));          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());      }  } |

**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>3.11.0</version>              <scope>test</scope>          </dependency>            <dependency>              <groupId>org.testng</groupId>              <artifactId>testng</artifactId>              <version>6.14.2</version>          </dependency>      </dependencies>  </project> |

**TestNG.xml**

TestNG.xml

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 | <?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" >      <listeners>          <listener class-name="com.InvokedMethodListener"></listener>      </listeners>      <test name="com.FirstTest" parallel="methods" thread-count="4">          <parameter name="browser" value="chrome"/>          <classes>              <class name="com.FirstTest">                  <methods>                      <include name="GOOGLE0" />                      <include name="GOOGLE2" />                      <include name="GOOGLE3" />                      <include name="GOOGLE4" />                  </methods>              </class>          </classes>      </test>      <test name="com.SecondTest"  parallel="methods" thread-count="2">          <parameter name="browser" value="chrome"/>          <classes>              <class name="com.SecondTest">                  <methods>                      <include name="GOOGLE1" />                      <include name="YANDEX" />                  </methods>              </class>          </classes>      </test>  </suite> |

**TestNG Parallel Execution Result for Above Implementation**

As you see below screenshot, all tests are run in parallel and passed.

