PARALLELTESTING

(SELENIUM JAVA+TESTNG)

*Submitted by*

*ANJANA P*

*BATCH CODE: 2023-11265*

ENROLLMENT NUMBER: EBEON0224885322

**INDEX**

|  |  |  |  |
| --- | --- | --- | --- |
| **1** | | **ACKNOWLEDGEMENT** | |
| **2** | | **ABSTRACT** | |
| **3** | | **INTRODUCTION** | |
| **4** | | REQUIREMENT ANALYSIS  4.1 OBJECTIVES  4.2 EXISTING SYSTEM | |
| **5** | | **PERFORM THE PROGRAM** | |
| **6** | | **CODE AND IMPLEMENTATION** | |
| **7** | **OUTPUTS** | |
| **8** | **CONS AND PROS** | |
| **9** | **CONCLUSION** | |

**ACKNOWLEDGEMENT**

I would like to express my heartfelt gratitude to all those who have contributed to the successful completion of this project report.

First and foremost, I extend my deepest appreciation to my project supervisor SANTHOSH KUMAR, for their unwavering support, guidance, and valuable insights throughout the entire project.

I am grateful to SANTHOSH KUMAR for providing valuable feedback and suggestions that have helped refine the contents of this report.

I would like to acknowledge the support and cooperation received from EDUBRIDGE as well as the resources and facilities provided, which have significantly contributed to the successful execution of this project.

Last but not least, I would like to express my deep appreciation to my family and friends for their understanding, encouragement, and patience throughout this project journey. Their unwavering support has been a constant source of motivation.

Each of the above individual’s contributions has played a crucial role in shaping this report and enhancing my learning experience. I am truly grateful for their involvement and support

**ABSTRACT**

Parallel testing is a process where multiple tests are execute simultaneously/in parallel in different thread processes. With respect to Selenium and TestNG. It allows you to execute multiple tests on different browsers, devices, environments in parallel and at the same time, instead of running it sequentially. TestNG provides multiple ways to execute tests in separate threads. In testng. xml, if we set 'parallel' attribute on the tag to 'tests', testNG will run all the '@Test' methods in tag in the same thread, but each tag will be in a separate thread. The main purpose of running tests in parallel mode is to reduce execution time and do maximum environment coverage (browsers/devices/environment) in less time.

**INTRODUCTION**

**Parallel testing or parallel execution, as the name suggests, is a process of running the test case parallel rather than one after the other.** In parallel testing, the program's multiple parts (or modules) execute together, saving the testers a lot of time and effort. The operating system's functionalities do this, but as a user, we need to trigger parallel execution through **[TestNG](https://www.toolsqa.com/testng/what-is-testng/)***.*As an example, you can think of having software with two different versions and running them in parallel with the help of TestNG. Parallel execution would give us the correct idea of the stability and performance of the software much faster than running serially*.* **TestNG** is a popular testing framework for Java that provides powerful features and advantages over other testing frameworks. It provides advanced features and functionalities for designing, organizing, and executing tests. TestNG offers a wide range of capabilities such **as parallel test execution, flexible test configuration, data-driven testing, test reporting**, and more. It supports a variety of test types including unit tests, integration tests, and end-to-end tests.

**Requirement analysis**

Parallel testing is a process where multiple tests are executed simultaneously/in parallel in different thread processes with respect to Selenium and TestNG. It allows you to execute multiple tests on different browsers, devices, environments in parallel and at the same time, instead of running it sequentially. The main purpose of running tests in parallel mode is to reduce execution time and do maximum environment coverage (browsers/devices/environment) in less time. TestNG has its rules too. Although it is reasonably evident that parallel testing must be used with the test case methods to run them in parallel TestNG offers three more areas where we can go ahead with parallel testing, combining these four areas, parallel testing accepts the following keywords (values) in TestNG:

* ***Methods***: This will run the parallel tests on all @Test methods in TestNG.
* **Tests**: All the test cases present inside the <test> tag will run with this value.
* **Classes**: All the test cases present inside the classes that exist in the XML will run in parallel.
* **Thread**-Count: Number of instances running to execute multiple tests parallel.
* **Suite**: collection of TestCases.

TestNG provides multiple ways to execute tests in separate threads. In testng. xml, if we set 'parallel' attribute on the tag to ‘tests’,testNG will run all the @Test methods in tag in the same thread,but each tag will be in a separate thread. we are now all set to run our first test case parallelly in TestNG using Selenium.

OBJECTIVES

The main Objective was to provide Parallel TestNG which provides:

**1. Reduces Time**

**2. Allow Multi-Threaded Tests**

**3. Improved efficiency**

EXISTING SYSTEM

In parallel testing, multiple tests can be run simultaneously in different execution modes, reducing execution time. This approach is particularly useful when running tests across multiple browsers or operating systems because it simplifies cross-browser testing. Parallel execution is a very important concept in the field of automated testing. All the work we do at the same time always saves time. Similarly, in end-to-end testing of an application, running tests in parallel instead of sequentially saves us completion time; This will save us more time in the remaining stages of the software testing lifecycle and application delivery.

Run parallel testing to improve our Selenium testing by running the most tests in the shortest amount of time. It helps large organizations run large-scale experiments. However, automation testers need to remember the following points before testing the same; for example, avoiding dependence of one test on another and adjusting parameters so that the test cannot be repeated successfully over several successful experiments.

Parallel tests using TestNG and Selenium are a powerful way to run multiple tests simultaneously, thereby reducing overall execution time.

**PERFORM THE PROGRAM**

**Step 1:**Create a Maven Project

**Step 2:**Add Dependencies

Open the pom.xml file in the project and add dependencies for TestNG and Selenium WebDriver.

**Step 3:**Set up a Java Project

If you are not using Maven you can set up a Java project by creating a new Java project and adding TestNG and Selenium WebDriver libraries to your IDE

**Step 4:**Create a Test Group

Now you can start creating a test group. You can organize your tests as **‘@Test’, ‘@BeforeTest’, ‘@AfterTest’**etc. Publish with TestNG annotations. and write Selenium WebDriver code in this process to run your tests.

**Step 5:** Write TestNG XML Configuration

Create a TestNG XML configuration file **(‘testng.xml’)**where you can define your test suites, test classes, and any parameters.

**Step 6:**Run Tests

**Step7:**View Results

After you complete the test, you can view the test results in the console output or in the HTML report generated by TestNG.

**CODE AND IMPLEMENTATION**

**package** paralleltestng;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.Keys;

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

**import** org.openqa.selenium.WebElement;

**import** org.openqa.selenium.chrome.ChromeDriver;

**import** org.testng.annotations.BeforeMethod;

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

**public** **class** ParallelTestngDemo

{

WebDriver driver;

@BeforeMethod

**public** **void** setUp()

{

//intialisation of chrome driver

driver = **new** ChromeDriver();

}

@Test

**public** **void** demoWebShop()

{

//navigate to the url

driver.get("https://demowebshop.tricentis.com/computers");

WebElement shoppingkart = driver.findElement(By.*className*("cart-label"));

shoppingkart.click();

}

@Test

**public** **void** openNon()

{

driver.get("https://demo.nopcommerce.com/login?returnUrl=%2F");

WebElement email = driver.findElement(By.*id*("Email"));

email.sendKeys("anju303123@gmail.com");

WebElement password = driver.findElement(By.*id*("Password"));

password.sendKeys("Demo1234" );

WebElement btnLogin = driver.findElement(By.*className*("login-button"));

btnLogin.click();

}

@Test

**public** **void** openShopsy()

{

driver.get("https://www.shopsy.in/");

WebElement searchbox = driver.findElement(By.*name*("q"));

searchbox.sendKeys("watches" + Keys.***ENTER***);

}

}

TESTNG.XML

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">

<suite name="Suite">

<test thread-count="3" name="Test" parallel = "methods">

<classes>

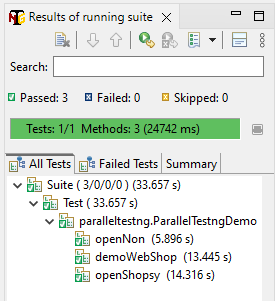
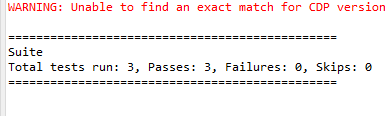
<class name="paralleltestng.ParallelTestngDemo"/>

</classes>

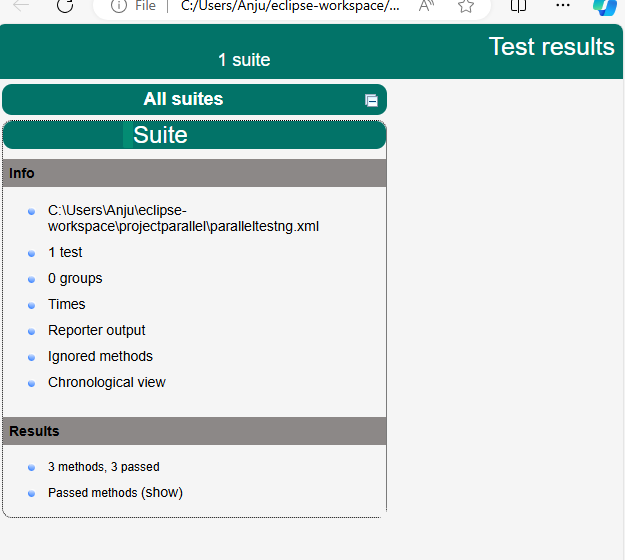
</test> <!-- Test -->

</suite> <!-- Suite -->

**OUTPUTS**



REPORT



**CONS AND PROS**

**PROS**

* Reduces Time: Running the tests in parallel reduces the overall execution time.
* Allow Multi-Threaded Tests: Using the parallel execution in TestNG, we can allow multiple threads to run simultaneously on the test case providing independence in the execution of different components of the software.
* Improved efficiency: Parallel testing maximizes resource utilization by distributing tests across multiple machines, threads, or processes. It improving testing efficiency.
* Early bug detection: Running tests in parallel allows for faster detection of bugs and issues. Defects can be identified and addressed earlier in the development cycle, leading to faster resolution and reduced overall costs.
* We are just organizing the execution by the use of annotation.
* Parallel testing is considerate while testing independent modules/classes. It fails for modules which are dependent on another module thereby giving flaky results.
* Tester should have a detailed understanding of the product under test and the workflow of the testcases to apply parallelism. If any dependent module is run in parallel mode, complete test execution may go for a toss.

**CONS**

* Fails On Dependent Modules : Most of the times the tests are inter-dependent, hence failing chances are more.
* Program Flow Sequence : The tester should be well aware of the program flow to create parallel testing modules.

**CONCLUSION**

We have probably mentioned all the scenarios that a software tester must handle during the testing phase of parallel testing. Integrated testing is possible when cross-browser testing is required; This requires installing Selenium Grid or accessing a cloud-based platform that provides you with Selenium Grid to run automated tests in parallel or in conjunction with different configurations. Integrated testing not only saves testers’ time, but also speeds up the delivery process following good results of verification and validation during testing. Try to improve your automated tests.