# Parameterization in TestNG using TestNG XML file

Parameterization is an important feature provided by [TestNG](https://www.scientecheasy.com/2019/04/testng-tutorial.html/) which allows the users to pass parameter values to test methods as arguments.

Using this feature, developers can run the same test over and over again with different values.

Parameterization test is supported by using Parameter and DataProvider annotations.

There are mainly two ways through which we can directly pass parameter values to test methods. They are:

1. Through [*testng.xml file*](https://www.scientecheasy.com/2020/07/create-run-testng-xml-file.html/)
2. Through DataProvider

In this tutorial, we will learn the first way and in the next tutorial, we will learn DataProvider.

## Parameterization through TestNG XML File

This approach is used to send parameter values (String types) through the testng.xml file directly to test methods at runtime. To use this technique, we have to use Parameters annotation for passing parameter values to test methods.

Let’s create a simple example program where we will pass parameter values to test methods through an XML configuration file. Follow the steps below:

1. Open Eclipse and create a new Java project with the following package structure.

2. Then, create a Java class file by the name “ParameterTestthroughXML” and copy the following code to it.

**Program source code 1:**

package parameterThrougXML;

import org.testng.annotations.Parameters;

import org.testng.annotations.Test;

public class ParameterTestthroughXML

{

// This test method will take one parameter as input. The value of parameter is defined at suite level.

@Parameters({"Name"})

@Test(priority = 1)

public void stName(String n)

{

System.out.println("Student's name: " +n);

}

@Parameters({"School"})

@Test(priority = 2)

public void scName(String sc)

{

System.out.println("School's name: " +sc);

}

// This test method will take two parameters as input.

// The value of test parameter is defined at the test level. The suite level parameter is overridden at test level.

@Parameters({"STD", "RollNo"})

@Test(priority = 3)

public void rollSTD(String s, String r)

{

System.out.println("STD: " +s);

System.out.println("Roll No: " +r);

}

}

In the preceding example program, we have created a test class with multiple test methods that accept a different set of parameter values.

As you can see in the program, Parameters annotation has been used for each test method with the name of parameter. The value of these parameters will be passed to the test method at the time of test execution by using the testng.xml file.

Let’s create a testng.xml file with name ParameterTest.xml and copy the following code to it.

<suite name="Suite" verbose="1">

<test name="Test">

<parameter name = "Name" value = "Deep"/>

<parameter name = "School" value = "RSVM" />

<parameter name = "STD" value = "Six" />

<parameter name = "RollNo" value = "04" />

<classes>

<class name = "parameterThrougXML.ParameterTestthroughXML"/>

</classes>

</test>

</suite>

In the preceding testng.xml file, the parameter is declared using a parameter tag. The attribute “name” of tag defines the name of the parameter whereas attribute “value” defines the value of said parameter. The tag can be used at suite level and test level in testng.xml file.

Now, run the preceding testng.xml file as TestNG test suite. You can see the following test result on the console window:

Output:

Student's name: Deep

School's name: RSVM

STD: Six

Roll No: 04

===============================================

Suite Total tests run: 3, Failures: 0, Skips: 0

===============================================

**Key points:**

1. The parameter values can be set at both suite and test level in the testng.xml file.

2. If you define parameter value at suite level then the parameter value set at test level will override it.

Let’s create a program where we will automate a Google home page by using the parameterization technique provided by TestNG and get the title of the homepage.

In this example, we will pass browser, and URL of Google home page as parameter values to the test methods from the testng.xml configuration file.

Look at the following source code to automate the above scenario.

**Program source code 2:**

package parameterThrougXML;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.testng.annotations.AfterTest;

import org.testng.annotations.BeforeTest;

import org.testng.annotations.Parameters;

import org.testng.annotations.Test;

public class ParameterTest

{

// Create first WebDriver reference.

WebDriver driver;

@Parameters({"Browser"})

@BeforeTest

public void setUp(String browser)

{

if(browser.equalsIgnoreCase("Firefox"))

{

driver = new FirefoxDriver();

System.out.println("Firefox opened");

}

else {

System.setProperty("webdriver.chrome.driver", "C:\chromedriver.exe");

driver = new ChromeDriver();

System.out.println("Chrome opened");

}

driver.manage().window().maximize();

}

@Parameters({"URL"})

@Test(priority = 1)

public void getURL(String URL)

{

driver.get(url);

}

@Test(priority = 2)

public void getTitle()

{

String title = driver.getTitle();

System.out.println("Title of Webpage: " +title);

}

@AfterTest

public void close()

{

driver.close();

}

}

Create a new testng.xml configuration file with name ParameterTest.xml and copy the following code to it.

<suite name="Suite" verbose="1">

<test name="Test">

<parameter name = "Browser" value = "Firefox"/>

<parameter name = "URL" value = "https://www.google.com"/>

<classes>

<class name = "parameterThrougXML.ParameterTest"/>

</classes>

</test>

</suite>

Now select the above testng.xml file and run it as TestNG suite. You will see the following results in the console window of Eclipse.

Output:

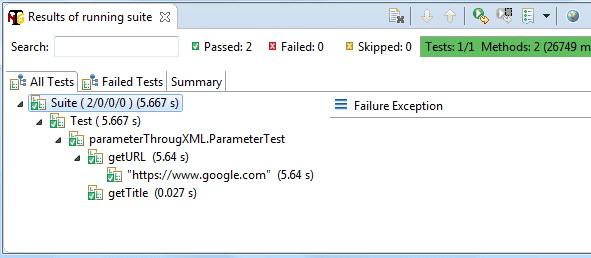
Firefox opened

Title of Webpage: Google

===============================================

Suite Total tests run: 2, Failures: 0, Skips: 0

===============================================

[](https://www.scientecheasy.com/2020/01/parameterization-in-testng.html/)

We have successfully created and run two example programs using the parameterization technique in TestNG.

**Key point:** The parameter annotation can be used with Before/After, Test, and Factory annotated methods.