**TestNG Parameters – Passing Values to Test Methods**

**What are Parameters in TestNG?**

In TestNG, **parameters** allow passing values from the TestNG XML file to test methods. This is useful when you need to run the same test with different inputs without modifying the test code.

**1. Using @Parameters Annotation**

* The @Parameters annotation helps pass values to test methods from the **TestNG XML file**.
* This is useful for **data-driven testing** where the same test case runs with different values.

**Example: Passing a Single Parameter**

**TestNG XML file (testng.xml)**

xml

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

<suite name="Test Suite">

<test name="Parameter Test">

<parameter name="username" value="testuser"/>

<classes>

<class name="ParameterExample"/>

</classes>

</test>

</suite>

**Java Test Case (ParameterExample.java)**

java

import org.testng.annotations.Parameters;

import org.testng.annotations.Test;

public class ParameterExample {

@Test

@Parameters("username")

public void testParameter(String username) {

System.out.println("The provided username is: " + username);

}

}

✅ **How it works?**

* The value "testuser" is passed from the XML file to the testParameter method.
* The test will print:

The provided username is: testuser

**2. Passing Multiple Parameters**

You can pass **multiple parameters** from XML and use them in a test method.

**TestNG XML file (testng.xml)**

<suite name="Test Suite">

<test name="Multiple Parameters Test">

<parameter name="username" value="admin"/>

<parameter name="password" value="admin123"/>

<classes>

<class name="MultiParameterExample"/>

</classes>

</test>

</suite>

**Java Test Case (MultiParameterExample.java)**

java

import org.testng.annotations.Parameters;

import org.testng.annotations.Test;

public class MultiParameterExample {

@Test

@Parameters({"username", "password"})

public void testMultipleParameters(String username, String password) {

System.out.println("Username: " + username);

System.out.println("Password: " + password);

}

}

✅ This will print:

Username: admin

Password: admin123

**3. Using Parameters in @BeforeClass or @BeforeMethod**

* You can also pass parameters to setup methods like @BeforeClass.
* This is useful for setting up configurations before test execution.

**Example: Parameter in @BeforeClass**

java

import org.testng.annotations.\*;

public class BeforeClassParameterExample {

@BeforeClass

@Parameters("browser")

public void setup(String browser) {

System.out.println("Test execution will run on: " + browser);

}

@Test

public void testMethod() {

System.out.println("Executing test...");

}

}

**TestNG XML file**

xml

<suite name="Suite">

<test name="Browser Test">

<parameter name="browser" value="Chrome"/>

<classes>

<class name="BeforeClassParameterExample"/>

</classes>

</test>

</suite>

✅ Output:

nginx

Test execution will run on: Chrome

Executing test...

**4. Difference Between @Parameters and @DataProvider**

| **Feature** | **@Parameters** | **@DataProvider** |
| --- | --- | --- |
| Source | XML file | Method inside the test class |
| Values | Hardcoded in XML | Can be dynamically generated |
| Flexibility | Less flexible | More flexible for multiple sets of data |
| Use Case | Simple parameter passing | Data-driven testing |

**When to Use @Parameters?**

✅ When you want **external configuration** from XML (e.g., browser type, environment URL).  
✅ When you need to **pass fixed values** to multiple test cases.  
✅ When test cases **don't need dynamic data generation**.