**TestNG @DataProvider – Data-Driven Testing**

**What is @DataProvider in TestNG?**

@DataProvider is a TestNG feature that allows us to pass multiple sets of data to a single test method. Unlike @Parameters, which takes values from XML, @DataProvider provides data from a method within the test class.

**1. Basic Example of @DataProvider**

java

import org.testng.annotations.DataProvider;

import org.testng.annotations.Test;

public class DataProviderExample {

@DataProvider(name = "loginData")

public Object[][] getData() {

return new Object[][] {

{"user1", "pass1"},

{"user2", "pass2"},

{"user3", "pass3"}

};

}

@Test(dataProvider = "loginData")

public void loginTest(String username, String password) {

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

}

}

✅ **How it works?**

* The getData() method returns a **2D array** containing multiple sets of login credentials.
* The test method loginTest() runs **three times**, once for each set of data.

📌 **Output:**

yaml

Username: user1, Password: pass1

Username: user2, Password: pass2

Username: user3, Password: pass3

**2. Using @DataProvider with Different Data Types**

* You can return any type of data, including int, boolean, or Objects.

java

@DataProvider(name = "numbers")

public Object[][] provideNumbers() {

return new Object[][] { {1}, {2}, {3} };

}

@Test(dataProvider = "numbers")

public void testNumbers(int num) {

System.out.println("Number: " + num);

}

📌 **Output:**

javascript

Number: 1

Number: 2

Number: 3

**3. DataProvider from Another Class**

You can store data in a **separate class** and use it in your test class.

**Data Provider Class**

java

import org.testng.annotations.DataProvider;

public class TestData {

@DataProvider(name = "userData")

public static Object[][] getUserData() {

return new Object[][] {

{"admin", "admin123"},

{"guest", "guest123"}

};

}

}

**Test Class**

java

import org.testng.annotations.Test;

public class ExternalDataProviderExample {

@Test(dataProvider = "userData", dataProviderClass = TestData.class)

public void testLogin(String username, String password) {

System.out.println("Testing with: " + username + " / " + password);

}

}

✅ **Benefit:**

* **Keeps test data separate** from the test logic.
* **Improves maintainability** of test cases.