**@DataProvider Annotation:**

The data provider is another annotation which supports data-driven testing.

i.e. To run the @Test method multiple times with different inputs.

We can use it to handle a broad range of complex parameters like the following.

* Java objects
* Objects from a database
* Data from Excel or property files

Specifying parameters in testng.xml might not be sufficient if we need to pass complex parameters like above.

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

<suite name="Parameters Testing Test Suite">

<test name="Parameters Testing">

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

<parameter name="UserName" value="demo"/>

<parameter name="Passcode" value="PASSWORD"/>

<classes>

<class name="com.techbeamers.testng.ParametersTesting" />

</classes>

</test>

</suite>

import org.testng.annotations.Parameters;

import org.testng.annotations.Test;

public class ParametersTesting {

@Parameters({ "BrowserName" })

@Test

public void OpenBrowser(String BrowserName) {

System.out.println("browser passed as :- " + BrowserName);

}

@Parameters({ "UserName", "Passcode" })

@Test

public void FillLoginForm(String UserName, String Passcode) {

System.out.println("Parameter for User Name passed as :- " + UserName);

System.out.println("Parameter for Passcode passed as :- " + Passcode);

}

}

A Data Provider is a method in our class that returns an array of objects. This method is annotated with @DataProvider.

**Facts on @DataProvider Annotation**

* This annotation has only one string attribute which is its name. If we don’t specify a name, then the method’s name serves as the default name.
* A data provider method prepares and should always return a two-dimensional array Object, (object [ ] [ ] - data set row and parameter value columns)
* A data-driven test would run once for each set of data specified by the data provider object.

Number of rows present in the array will be the number of iteration of the @Test

Number of columns present in the array will be the number of arguments of the @Test

**How to Use @DataProvider Annotation**

To use the DataProvider feature in the tests, we have to declare a method annotated by @DataProvider and then use the method in the test method using the ‘dataProvider‘ attribute in the @Test annotation.

**TestNG @DataProvider and @Test in same class**

**import** java.util.concurrent.TimeUnit;

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

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

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

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

**import** org.testng.annotations.DataProvider;

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

**public** **class** DataProviderDemoClass {

@DataProvider(name="urlData")

**public** Object[][] searchData()

{

//single argument from the table

Object[][] data=**new** Object[3][1];

data[0][0]="https://google.com";

data[1][0]="https://yahoo.com";

data[2][0]="https://bing.com";

**return** data;

}

@Test(dataProvider="urlData")

**public** **void** doSearch(String site)

{

// To invoke browser

System.*setProperty*("webdriver.chrome.driver", "G:\\Selenium RS 2019\\chromedriver\_win32\\chromedriver.exe");

// Create object for browser driver

WebDriver driver = **new** ChromeDriver();

// Maximize window

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

driver.get(site);

driver.manage().timeouts().implicitlyWait(10, TimeUnit.***SECONDS***);

driver.close();

}

}

The test result for the respective test method in the class was executed 3 times. The execution of the test method is dependent upon the number of data-sets passed by the DataProvider method; in this case as three different sets of data were returned by the DataProvider, thus the test method was executed three times.

@DataProvider(name="usernamepwd")

**public** Object[][] loginData()

{

Object[][] data=**new** Object[3][2];

data[0][0]="mercury";

data[0][1]="mercury";

data[1][0]="test";

data[1][1]="test";

data[2][0]="admin";

data[2][1]="admin";

**return** data;

}

@Test(dataProvider="usernamepwd")

**public** **void** doLogin(String username,String password) **throws** InterruptedException

{

System.*setProperty*("webdriver.chrome.driver", "G:\\Selenium RS 2019\\chromedriver\_win32\\chromedriver.exe");

WebDriver driver = **new** ChromeDriver();

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

driver.get("http://newtours.demoaut.com/");

WebElement user = driver.findElement(By.*name*("userName"));

user.sendKeys(username);

WebElement pwd = driver.findElement(By.*name*("password"));

pwd.sendKeys(username);

WebElement signin = driver.findElement(By.*name*("login"));

signin.click();

driver.manage().timeouts().implicitlyWait(10, TimeUnit.***SECONDS***);

driver.quit();

}

**Limitations of the Data Provider:**

1. We cannot increase the array size as we know it from java itself

Example: Object [2][2] is fixed

1. We cannot use this for large set of data because it increases the program size
2. If the data is often changing, then we cannot use this data provider because it requires recompilation
3. We cannot use data from another class data provider

**Reference links:**

<https://chercher.tech/java/data-provider-testng-selenium-webdriver>

<https://howtodoinjava.com/testng/testng-dataprovider/>

<https://www.techbeamers.com/testng-parameters-and-dataprovider-annotations/#dataprovider-annotation>

<https://www.techbeamers.com/testng-parameters-and-dataprovider-annotations/>

<https://www.edureka.co/blog/dataprovider-in-testng/>

<https://www.softwaretestingmaterial.com/testng-parameterization-using-dataproviders/>

<https://www.toolsqa.com/selenium-webdriver/testng-data-provider-excel/>

<https://sqlandsiva.blogspot.com/2010/12/testng-passing-multiple-parameters-in.html>

<https://www.seleniumeasy.com/testng-tutorials/dataprovider-in-testng>

<https://stackoverflow.com/questions/16292434/how-to-pass-10-parameters-using-testng-dataprovider>

<https://examples.javacodegeeks.com/enterprise-java/testng/testng-dataprovider-example/>

<https://www.journaldev.com/21257/testng-dataprovider>

<https://www.javarticles.com/2015/03/example-of-testng-dataprovider.html>

<http://www.allinoneblogs.com/selenium-tutorials/testng/dataprovider-with-multiple-parameters/>

<https://www.lambdatest.com/blog/parameterization-in-testng-dataprovider-and-testng-xml-examples/>

**import** java.util.concurrent.TimeUnit;

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

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

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

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

**import** org.testng.annotations.DataProvider;

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

**public** **class** DataProviderDemoClass {

@DataProvider(name="urlData")

**public** Object[][] searchData()

{

//single argument from the table

Object[][] data=**new** Object[3][1];

data[0][0]="https://google.com";

data[1][0]="https://yahoo.com";

data[2][0]="https://bing.com";

**return** data;

}

@DataProvider(name="usernamepwd")

**public** Object[][] loginData()

{

Object[][] data=**new** Object[3][2];

data[0][0]="mercury";

data[0][1]="mercury";

data[1][0]="test";

data[1][1]="test";

data[2][0]="admin";

data[2][1]="admin";

**return** data;

}

@Test(dataProvider="urlData", priority = 1)

**public** **void** doSearch(String site)

{

// To invoke browser

System.*setProperty*("webdriver.chrome.driver", "G:\\Selenium RS 2019\\chromedriver\_win32\\chromedriver.exe");

// Create object for browser driver

WebDriver driver = **new** ChromeDriver();

// Maximize window

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

driver.get(site);

driver.manage().timeouts().implicitlyWait(10, TimeUnit.***SECONDS***);

driver.close();

}

@Test(dataProvider="usernamepwd", priority = 2)

**public** **void** doLogin(String username,String password) **throws** InterruptedException

{

System.*setProperty*("webdriver.chrome.driver", "G:\\Selenium RS 2019\\chromedriver\_win32\\chromedriver.exe");

WebDriver driver = **new** ChromeDriver();

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

driver.get("http://newtours.demoaut.com/");

WebElement user = driver.findElement(By.*name*("userName"));

user.sendKeys(username);

WebElement pwd = driver.findElement(By.*name*("password"));

pwd.sendKeys(username);

WebElement signin = driver.findElement(By.*name*("login"));

signin.click();

driver.manage().timeouts().implicitlyWait(10, TimeUnit.***SECONDS***);

driver.quit();

}

}