Test NG

Class 4

alwaysRun

For before methods (beforeSuite, beforeTest, beforeTestClass and beforeTestMethod, but not beforeGroups): If set to true, this configuration method will be run regardless of what groups it belongs to. For after methods (afterSuite, afterClass, ...): If set to true, this configuration method will be run even if one or more methods invoked previously failed or was skipped.

Task

Identify Priority of Test Cases

TC 1: Saucedemo Username1(tag the groups - Smoke)
Step 1: Open browser and navigate to Saucedemo
Step 2: Enter username standard_user and enter password secret_sauce and click login button
Step 3: Verify user successfully logged in

TC 2: Saucedemo Username2(tag the groups - Regression)
Step 1: Open browser and navigate to Saucedemo
Step 2: Enter username problem_user and enter password secret_sauce and click login button
Step 3: Verify user successfully logged in

Note: Create BeforeMethod and AfterMethod annotations to open browser and logout from the application. Create a xml file and include smoke.

Agenda

Test Parameterization in TestNG

Parameters in TestNG

DataProvider in TestNG

TestNG Attributes

Attribute	Description
priority	The priority for this test method.
enabled	Sets whether the said method or the methods inside the said class should be enabled for execution or not. By default its value is true.
groups	List of groups the said method or class belongs to.
dependsOnGroups	Specifies the list of groups this method depends on.
dependsOnMethods	Specifies the list of methods this method depends on.
description	The description of this method.
dataProvider	The name of the data provider, which will provide data for data-driven testing to this method.

Test Parameterization in TestNG

Test case parameterization is the technique to execute the same Test Case with different input values. Parameterized tests allow us to run the same tests over and over again using different values.

It allows us to automatically run a test case multiple times with different input and validation values.

There are two ways in TestNG to make the test case parameterized test.

1) Using **@Parameters + declaring parameters in testng.xml.**
2) Using **@DataProvider**

@Parameters Annotation in TestNG

TestNG allows user to pass values to test methods as arguments by using @Parameters annotation through testng.xml file.

This technique of parameterization can be used when the scope of Test data is limited. Suppose, we want to pass Browser Name run time i.e on which browser test will run. So, in this case, we can use this kind of parameterization instead of hardcode the values in our code.

Actually, it functions on the concept of key-value pair. We declare the key within the @Parameters({“key”}) annotations after @Test annotation and declare the values in testng.xml file.

@Parameters Annotation in TestNG

```
import org.testng.annotations.Parameters;
import org.testng.annotations.Test;

public class TestParameters {

	@Parameters({ "browser" })
	@Test
	public void testCaseOne(String browser) {
		SOP("browser passed as :- " + browser);
	}

	@Parameters({ "username", "password" })
	@Test
	public void testCaseTwo(String username, String password) {
		SOP("Parameter for User Name passed as :- " + username);
		SOP("Parameter for Password passed as :- " + password);
	}
}
```

Parameters in testng.xml

```
<suite name="Parameterization Test Suite">
	<test name="Testing Parameterization">
	<parameter name="browser" value="Firefox"/>
	<parameter name="username" value="testuser"/>
	<parameter name="password" value="testpassword"/>
		<classes>
			<class
name="com.parameterization.TestParameters" />
		</classes>
	</test>
</suite>
```

Task

Identify Priority of Test Cases

TC 1: Saucedemo Username1(parameters - username and password)

Step 1: Open browser and navigate to Saucedemo

Step 2: Enter username standard_user and enter password secret_sauce and click login button

Step 3: Verify user successfully logged in

TC 2: Saucedemo Username2(parameters - username and password)

Step 1: Open browser and navigate to Saucedemo

Step 2: Enter username problem_user and enter password secret_sauce and click login button

Step 3: Verify user successfully logged in

Note: Create BeforeMethod and AfterMethod annotations to open browser and logout from the application. Create a xml file with parameters username and password .

@DataProvider in TestNG

When we want to pass complex parameters,
then @DataProvider is the way to go.

@DataProvider Annotation of testng framework provides us a facility of storing and preparing data set In method.
Task of @DataProvider annotated method Is supplying data for a test method.

Means you can configure data set In that method and then use that data In your test method. @DataProvider annotated method must return an Object[][] with data.

Parameterized Test Using @DataProvider

- We can use Data Providers when we want read test data from Database, Excel file, XML file or any another medium. Data Provider is a method annotated with @DataProvider. **A Data Provider returns an array of objects**. So the return type of DataProvider method must be two-dimensional object array.

  Steps to parameterize the Test:

- Create the data provider method with@DataProvider annotations. (Methods return type must be two-dimensional object arrays)

- Declare the “Data Provider” name after the @Test annotation. Like – @Test (dataProvider = “data provider method name”)

- Pass the arguments in the Test method, what you want to use from the data provider.

Interesting facts about the DataProvider

This annotation has 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 returns a 2D list of objects.

A data-driven test would run once for each set of data specified by the data provider object.

Number of values passed by a DataProvider method should be equal to number of attributes of test method on which DataProvider method is used. If DataProvider methods provided three attributes, test method must accept three attributes and vise versa.

We can execute a test method for multiple set of data using DataProvider which is not possible though Parameters annotation where we pass values from testng.xml.

@DataProvider in TestNG

```
@DataProvider  // Declare Data Provider Method
public Object[][] getData() {
      /*Rows - 3, test will repeat 3 times Columns - 2 , test will
      process * for 2 values  */
       Object[][] testData = new Object[3][2];
      // First Row
       testData[0][0] = "Selenium";
       testData[0][1] = "Selenium@";
      // second Row
       testData[1][0] = "Merilla";
       testData[1][1] = "Selenium@";
      // third Row
       testData[2][0] = "Eden";
       testData[2][1] = "Selenium@";
      }
}
```

@DataProvider in TestNG

```
 public class LoginParameterized {

// Declare Data Provider for Test Method
 @Test(dataProvider = "getData")
 public void loginAccount(String userName, String password) {
    SOP("Username is --> " + userName  +  "  Password is --> " +
password);
}
```