**TestNG Tutorial**

**packages**

Testng tutorial we can add the package ‘s also

<suite>

<test>

<packages>

<package name=com.test/>

</packages>

</test>

</suite>

This will consider will retain all the classes that has TestNG notation

**To allow the Test to return the value**

est methods are annotated with @Test. Methods annotated with @Test that happen to return a value will be ignored, unless you set allow-return-values to true in your testng.xml:

|  |
| --- |
| <suite allow-return-values="true"> |

|  |
| --- |
|  |

|  |
| --- |
| or |

|  |
| --- |
|  |

|  |
| --- |
| <test allow-return-values="true"> |
| **Groups of Groups**   |  | | --- | | test name="Regression1"> |  |  | | --- | | <groups> |  |  | | --- | | <define name="functest"> |  |  | | --- | | <include name="windows"/> |  |  | | --- | | <include name="linux"/> |  |  | | --- | | </define> |  |  | | --- | |  |  |  | | --- | | <define name="all"> |  |  | | --- | | <include name="functest"/> |  |  | | --- | | <include name="checkintest"/> |  |  | | --- | | </define> |  |  | | --- | |  |  |  | | --- | | <run> |  |  | | --- | | <include name="all"/> |  |  | | --- | | </run> |  |  | | --- | | </groups> |  |  | | --- | |  |  |  | | --- | | <classes> |  |  | | --- | | <class name="test.sample.Test1"/> |  |  | | --- | | </classes> |  |  | | --- | | </test> | |

**Parameters in Test ng**

<suite>

<paramtere name=”xyz” value=”abc”>

<test>

<classes>

<class name=org.prasanna.temp>

</classes>

</test>

</suite>

@prameter({“xyz”})

@Test

Public void test(String parm1)

{

Sop(param1)

}

**Optional parameter**

<suite>

<paramtere name=”xyz” value=”abc”>

<test>

<classes>

<class name=org.prasanna.temp>

</classes>

</test>

</suite>

Public void (@optional (“prasanna”) String pram1 )

{ sop(parm1)

}

On any method that already has a @Test, @Before/After or @Factory annotation

DataProvider in TestNG

[1.@Dataprovider(name=”prasanna”)](mailto:1.@Dataprovider(name=”prasanna”))

Public object[][] givedata()

{

Return new object[][]{{“prasanna”,”Testing testng”},{“MKV”,”TestNG”}};

}

@Test(dataprovider=”prasanna”)

Public void testdataprovider(String s1,String s2)

{

Sop(s1+s2)

}

--By default, the data provider will be looked for in the current test class or one of its base classes

If you want to put your data provider in a different class, it needs to be a static method and you specify the class where it can be found in thedataProviderClass attribute:

|  |
| --- |
| public class StaticProvider { |

|  |
| --- |
| @DataProvider(name = "create") |

|  |
| --- |
| public static Object[][] createData() { |

|  |
| --- |
| return new Object[][] { |

|  |
| --- |
| new Object[] { new Integer(42) } |

|  |
| --- |
| } |

|  |
| --- |
| } |

|  |
| --- |
| } |

|  |
| --- |
|  |

|  |
| --- |
| public class MyTest { |

|  |
| --- |
| @Test(dataProvider = "create", dataProviderClass = StaticProvider.class) |

|  |
| --- |
| public void test(Integer n) { |

|  |
| --- |
| // ... |

|  |
| --- |
| } |

|  |
| --- |
| } |
| **Dependencies with annotations** |