Extensions

**package** junitTestScripts;

**import** org.junit.jupiter.api.Test;

**import** org.junit.jupiter.api.condition.EnabledForJreRange;

**import** org.junit.jupiter.api.condition.EnabledOnJre;

**import** org.junit.jupiter.api.condition.EnabledOnOs;

**import** org.junit.jupiter.api.condition.JRE;

**import** org.junit.jupiter.api.condition.OS;

**public** **class** ExtensionsDemoCondition {

// test the method based on the following condition

/\*

  \*  > condition OS

  \*  > Condition based on JRE

  \*  > Condition based on range of Java

  \*

  \*

  \*  \*/

@Test

@EnabledOnOs(OS.***MAC***)

// condition is If the OS is Mac -> run the test else ignore/disable the test

**public** **void** testConditionOS()

{

System.***out***.println("OS is matching and test is exeucted");

}

@Test

@EnabledOnJre(JRE.***JAVA\_14***)

// condition is If the java version on laptop is 14 -> run the test else ignore/disable the test

**public** **void** testConditionJRE()

{

System.***out***.println("Java version is matching and test is exeucted");

}

@Test

@EnabledForJreRange(min = JRE.***JAVA\_10***, max= JRE.***JAVA\_17***)

// condition is If the java version on laptop is in between 10 to 17 -> run the test else ignore/disable the test

**public** **void** testConditionJRErange()

{

System.***out***.println("Java version is matching and test is exeucted");

}

}

@Test

@EnabledIfSystemProperty(named="java.vm.vendor", matches="Oracle.\*")

// condition is If the java version is provided by oracle-> run the test else ignore/disable the test

**public** **void** testConditionSystemProperty()

{

System.***out***.println("Java version is installed by oracle and test is exeucted");

}

 @Test

  @EnabledIfEnvironmentVariable(named = "myENVOS", matches="Linux") //

**public** **void** testConditionEnvVariables() {

  System.***out***.println(" Envirnoment varibale is linux and test is exeucted");

  }

/=============================================================/

Tagged test cases @Tag

**package** junitTestScripts;

**import** org.junit.jupiter.api.Tag;

**import** org.junit.jupiter.api.Test;

**import** org.junit.platform.suite.api.ExcludeTags;

**import** org.junit.runner.RunWith;

**import** org.junit.runners.Suite;

@RunWith(Suite.**class**)

**public** **class** IncludeExcludeTags {

@Test

@Tag("feature1")

**public** **void** feature1Test1()

{

System.***out***.println("test 1 for feature1");

}

@Test

@Tag("feature1")

**public** **void** feature1Test2()

{

System.***out***.println("test 2 for feature1");

}

@Test

@Tag("feature1")

**public** **void** feature1Test3()

{

System.***out***.println("test 3 for feature1");

}

@Test

@Tag("feature1")

**public** **void** feature1Test4()

{

System.***out***.println("test 4 for feature1");

}

@Test

@Tag("feature2")

**public** **void** feature2Test1()

{

System.***out***.println("test 1 for feature2");

}

@Test

@Tag("feature2")

**public** **void** feature2Test2()

{

System.***out***.println("test 2 for feature2");

}

@Test

@Tag("feature2")

**public** **void** feature2Test3()

{

System.***out***.println("test 3 for feature2");

}

}