System Property Conditions - @DisabledIfSystemProperty and @EnabledIfSystemProperty

This lesson demonstrates how to disable or enable test methods or a complete test class using System property level conditions.

WE'LL COVER THE FOLLOWING

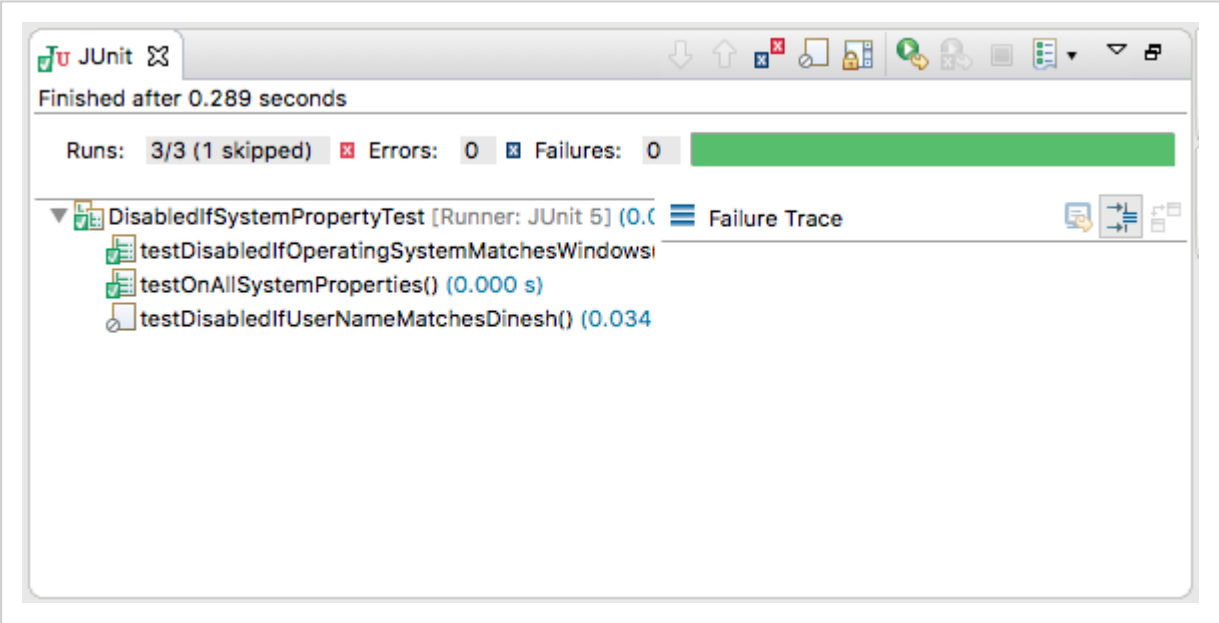
@DisabledIfSystemProperty and @EnabledIfSystemProperty

Junit 5 helps us to disable or enable test cases using various conditions. JUnit Jupiter API provides annotations in `org.junit.jupiter.api.condition` package to enable/disable tests based on a certain condition. The annotations provided by API can be applied to test methods as well as the class itself. The two annotations which use system properties and specified regex to disable or enable tests are - `@DisabledIfSystemProperty` and `@EnabledIfSystemProperty`. Let's take a look at a demo.

DisabledIfSystemPropertyTest.java

```
package com.hubberspot.junit5.disabled;

import static org.junit.jupiter.api.Assertions.assertFalse;
import static org.junit.jupiter.api.Assertions.assertTrue;

import org.junit.jupiter.api.Test;
import org.junit.jupiter.api.condition.DisabledIfSystemProperty;

public class DisabledIfSystemPropertyTest {

    @Test
    void testOnAllSystemProperties() {
        assertTrue(3 > 0);
    }
```

```
    @DisabledIfSystemProperty(named="user.name", matches="dinesh")
    @Test
    void testDisabledIfUserNameMatchesDinesh() {
        assertFalse(0 > 4);
    }

    @DisabledIfSystemProperty(named="os.name", matches="Windows")
    @Test
    void testDisabledIfOperatingSystemMatchesWindows() {
        assertFalse(10 > 40);
    }
}
```



Above test program has 3 test methods and `@DisabledIfSystemProperty` is applied on 2 test methods as.

1. `testDisabledIfUserNameMatchesDinesh()` - Here, `@DisabledIfSystemProperty` annotation takes in two attributes such as ***`named`*** and ***`matches`***. The value provided to ***`named`*** attribute is **`"user.name"`** and value provided to ***`matches`*** attribute is **`"dinesh"`**. It makes the test method skip to execute if System property by name `"user.name"` matches `"dinesh"`.
2. `testDisabledIfOperatingSystemMatchesWindows()` - Here, `@DisabledIfSystemProperty` annotation takes in two attributes such as ***`named`*** and ***`matches`***. The value provided to ***`named`*** attribute is **`"os.name"`** and value provided to ***`matches`*** attribute is **`"Windows"`**. It makes the test method skip to execute if System property by name `"os.name"` matches `"Windows"`.

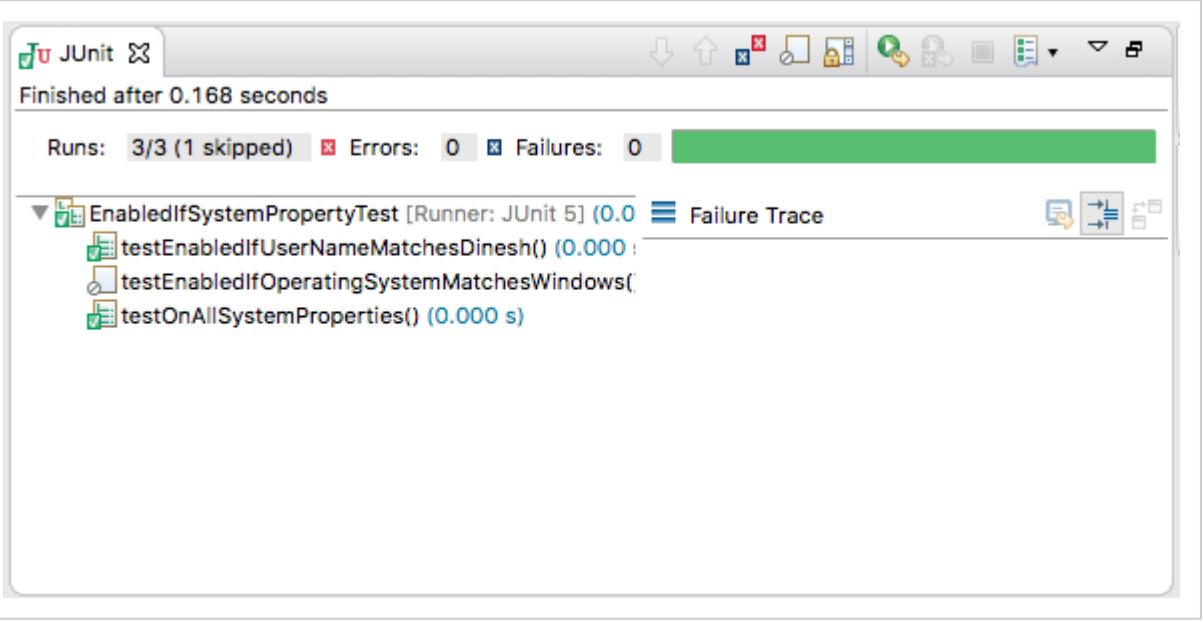
The above test methods are executed on System properties as -

1. `user.name` - `dinesh`
2. `os.name` - `Mac OS X`

Thus, output shows that 1 test method marked as, `@DisabledIfSystemProperty(named="user.name", matches="dinesh")` is skipped for execution.

EnabledIfSystemPropertyTest.java

```
package com.hubberspot.junit5.disabled;

import static org.junit.jupiter.api.Assertions.assertFalse;
import static org.junit.jupiter.api.Assertions.assertTrue;

import org.junit.jupiter.api.Test;
import org.junit.jupiter.api.condition.EnabledIfSystemProperty;

public class EnabledIfSystemPropertyTest {

    @Test
    void testOnAllSystemProperties() {
        assertTrue(3 > 0);
    }

    @EnabledIfSystemProperty(named="user.name", matches="dinesh")
    @Test
    void testEnabledIfUserNameMatchesDinesh() {
        assertFalse(0 > 4);
    }

    @EnabledIfSystemProperty(named="os.name", matches="Windows")
    @Test
    void testEnabledIfOperatingSystemMatchesWindows() {
        assertFalse(10 > 40);
    }
}
```



Output of tests

Above test program has 3 test methods and `@EnabledIfSystemProperty` is applied on 2 test methods as -

1. `testEnabledIfUserNameMatchesDinesh()` - Here, `@EnabledIfSystemProperty` annotation takes in two attributes such as ***`named`*** and ***`matches`***. The value provided to ***`named`*** attribute is **`"user.name"`** and value provided to ***`matches`*** attribute is **`"dinesh"`**. It makes the test method enable to execute if System property by name `"user.name"` matches `"dinesh"`.

2. `testEnabledIfOperatingSystemMatchesWindows()` - Here, `@EnabledIfSystemProperty` annotation takes in two attributes such as ***`named`*** and ***`matches`***. The value provided to ***`named`*** attribute is **`"os.name"`** and value provided to ***`matches`*** attribute is **`"Windows"`**. It makes the test method enable to execute if System property by name `"os.name"` matches `"Windows"`.

The above test methods are executed on System properties as -

1. **`user.name`** - **`dinesh`**
2. **`os.name`** - **`Mac OS X`**

Thus, output shows that 1 test method marked as, `@EnabledIfSystemProperty(named="os.name", matches="Windows")` is skipped for execution.