

Testing using UFT Pro (LeanFT) and JUnit in Java

Overview

In this tutorial, we will create a JUnit Test in Java using [UFT Pro \(LeanFT\)](#) from Micro Focus for browser automation.

 Learn more



[UFT Pro \(LeanFT\)](#) is an advanced functional test automation solution that was designed specifically for continuous integration and continuous testing. It provides tools and capabilities that easily and efficiently create robust test automation in the developer IDE and integrate it seamlessly into the CI process. LeanFT enables automating applications for a wide range of technologies, including web, mobile and desktop.

Description

The following automated test uses LeanFT library in order to navigate through a website and validates the price shown for a product versus the one presented when it was added to the shopping cart.

```
package testing;

import static org.junit.Assert.*;

import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Test;

import com.hp.lft.sdk.*;
import com.hp.lft.sdk.web.*;
import com.hp.lft.verifications.*;

import unittesting.*;

public class LeanFtTest extends UnitTestClassBase {

    public LeanFtTest() {
        //Change this constructor to private if you supply your own public constructor
    }

    @BeforeClass
    public static void setUpBeforeClass() throws Exception {
        instance = new LeanFtTest();
        globalSetup(LeanFtTest.class);
    }

    @AfterClass
    public static void tearDownAfterClass() throws Exception {
        globalTearDown();
    }
}
```

```

@Before
public void setUp() throws Exception {
}

@After
public void tearDown() throws Exception {
}

@Test
public void totalPriceTest() throws GeneralLeanFtException, InterruptedException {

    //Launch Chrome and navigate to the online store application
    Browser browser = BrowserFactory.launch(BrowserType.CHROME);
    browser.navigate("http://www.advantageonlineshopping.com");

    //Click the "Tablets" category
    Link tabletsLink = browser.describe(Link.class, new LinkDescription.Builder()
        .id("TabletsImg").build());
    tabletsLink.setDisplayName("Tablets");
    tabletsLink.click();

    //Click a specific tablet
    Image tabletElitePad = browser.describe(Image.class, new ImageDescription.Builder()
        .src("http://www.advantageonlineshopping.com/catalog/fetchImage?image_id=3100")
        .className("imgProduct").build());
    tabletElitePad.setDisplayName("Tablet ElitePad");
    tabletElitePad.click();

    //Add it to the cart
    browser.describe(Button.class, new ButtonDescription.Builder()
        .name("ADD TO CART").build()).click();

    //Store its price
    String tabletPrice = browser.describe(WebElement.class, new WebElementDescription.Builder()
        .className("roboto-medium cart-total ng-binding").build()).getInnerText();

    //Check out
    browser.describe(Button.class, new ButtonDescription.Builder()
        .className("roboto-medium ng-binding").build()).click();

    //Verify that the total price presented in the purchase summary page, is exactly the price of the
selected tablet
    String totalPrice = browser.describe(WebElement.class, new WebElementDescription.Builder()
        .className("roboto-medium totalValue ng-binding")
        .innerText(new RegExpProperty("\\$.*")).build()).getInnerText();

    Verify.areEqual(tabletPrice, totalPrice, "Verify total price");
}
}

```

After running the Test Case successfully and generating the JUnit XML report (e.g., [TEST-testing.LeanFtTest-success.xml](#)), it can be imported to Xray (by using either the REST API or the **Import Execution Results** action within the Test Execution).

Description

Execution for a successful use case.

Tests

Overall Execution Status

TOTAL TESTS: 1

1 PASSED

Key	Summary	Test Type	Status	Actions
☐ CALC-2	totalPriceTest	Generic	PASSED	

Prev 1 Next

Total 1 issues

Test Environments

Associated Test Environments

None

Assignee: Unassigned
[Assign to me](#)

Reporter: Sousa

Votes: 0

Watchers: 1 [Stop watching this issue](#)

Created: 1 minute ago

Updated: Just now

Agile

[View on Board](#)

JUnit's Test Case is mapped to a Generic Test in Jira, and the **Generic Test Definition** field contains the package name, followed by the name of the class and the method name that implements the Test Case.

The Execution Details of the Generic Test contains information about the Test Suite, which in this case corresponds to the package name, followed by the Test Case class.

Execution Status PASSED

Started On: 26/Jun/2018 1:31 PM

Finished On: 26/Jun/2018 1:31 PM

Assignee: Versions: -

Unassigned Revision: -

Executed By: Sousa

Test Environments: -

Comment

[Preview comment](#)

Execution Defects (0)

[Add Defects](#)

[Create Defect](#)

Execution Evidences (0)

[Add Evidences](#)

Execution Details

Test Description

None

Test Details

Test Type: Generic
Definition: testing.LeanPtTest.totalPriceTest

Results

Context	Error Message	Duration	Status
TestSuite testing.LeanPtTest	-	34 secs	PASSED

If the Test fails, for example, due to a missing web element (e.g., [TEST-testing.LeanPtTest-fail.xml](#)), then besides the overall Test Run being marked as FAIL, you can also see detailed information on the exception that was raised during the execution of the automated test.

Execution Status ❌ FAILED

Started On: 26/Jun/2018 10:19 AM

Finished On: 26/Jun/2018 10:19 AM

Assignee:

Unassigned

Executed By: Sousa

Test Environments: -

Versions: -

Revision: -

Comment Preview comment

Execution Defects (0) Add Defects Create Defect

Execution Evidences (0) Add Evidences

Execution Details

Test Description

None

Test Details

Test Type: Generic
Definition: testing.LeanFTTest.totalPriceTest

Results

Context	Error Message	Duration	Status
TestSuite testing.LeanFTTest	<pre>com.hp.lft.sdk.ReplayObjectNotFoundException: Cannot identify the object "Web Button". Verify that this object's properties match an object currently displayed in your application. at com.hp.lft.sdk.internal.ReplayExceptionFactory\$1.create(ReplayExceptionFactory.java:34) at com.hp.lft.sdk.internal.ReplayExceptionFactory.createOrDefault(ReplayExceptionFactory.java:197) at com.hp.lft.sdk.internal.ReplayExceptionFactory.createOrDefault(ReplayExceptionFactory.java:21) at com.hp.lft.sdk.internal.TestObjectExecutorBehaviorBase\$ReplayErrorHandler.onError(TestObjectExecutorBehaviorBase.java:65) at com.hp.lft.sdk.internal.CommunicationClientImpl.handleError(CommunicationClientImpl.java:221) at com.hp.lft.sdk.internal.CommunicationClientImpl.send(CommunicationClientImpl.java:96) at com.hp.lft.sdk.internal.TestObjectExecutorBehavior.executeMethod(TestObjectExecutor.java:33) at com.hp.lft.sdk.internal.TestObjectBase.executeMethod(TestObjectBase.java:119) at com.hp.lft.sdk.internal.ClickBehavior.click(ClickBehavior.java:37) at com.hp.lft.sdk.internal.ClickBehavior.click(ClickBehavior.java:32) at com.hp.lft.sdk.internal.web.WebElementImpl\$1.invoke(WebElementImpl.java:29) at com.hp.lft.sdk.internal.TestObjectOperationWrapper.executeWithEvents(TestObjectOperationWrapper.java:81) at com.hp.lft.sdk.internal.TestObjectOperationWrapper.executeWithEvents(TestObjectOperationWrapper.java:102) at com.hp.lft.sdk.internal.web.WebElementImpl.click(WebElementImpl.java:33) at testing.LeanFTTest.totalPriceTest(LeanFTTest.java:72)</pre>	34 secs	FAILED

Note that if you're using LeanFT's "Verify" method, that verification won't raise an exception by itself and thus, the Test will appear as passed (if it didn't fail until then) even if the verification itself failed.

The "Verify" class methods return a Boolean value reflecting the verification result. If it is "false", it is possible to manually throw an exception to make the test status reflect the actual verification result.

References

- <https://software.microfocus.com/en-us/software/leanft>