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.

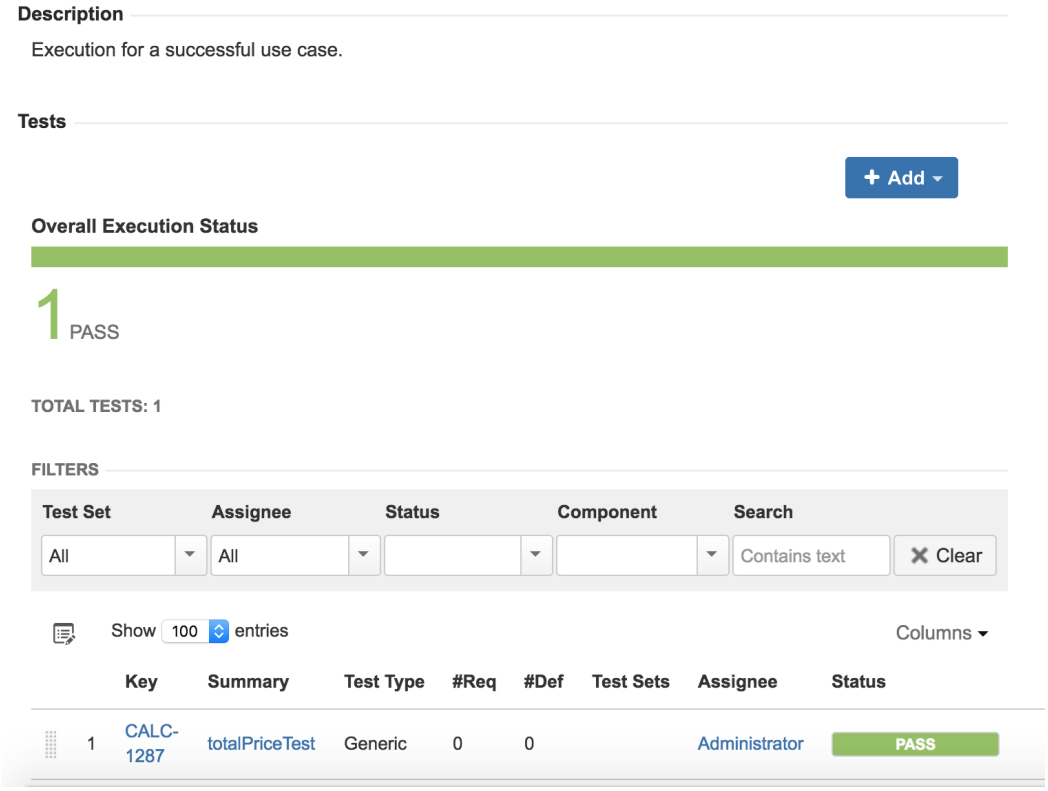
Description

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();
    }
```

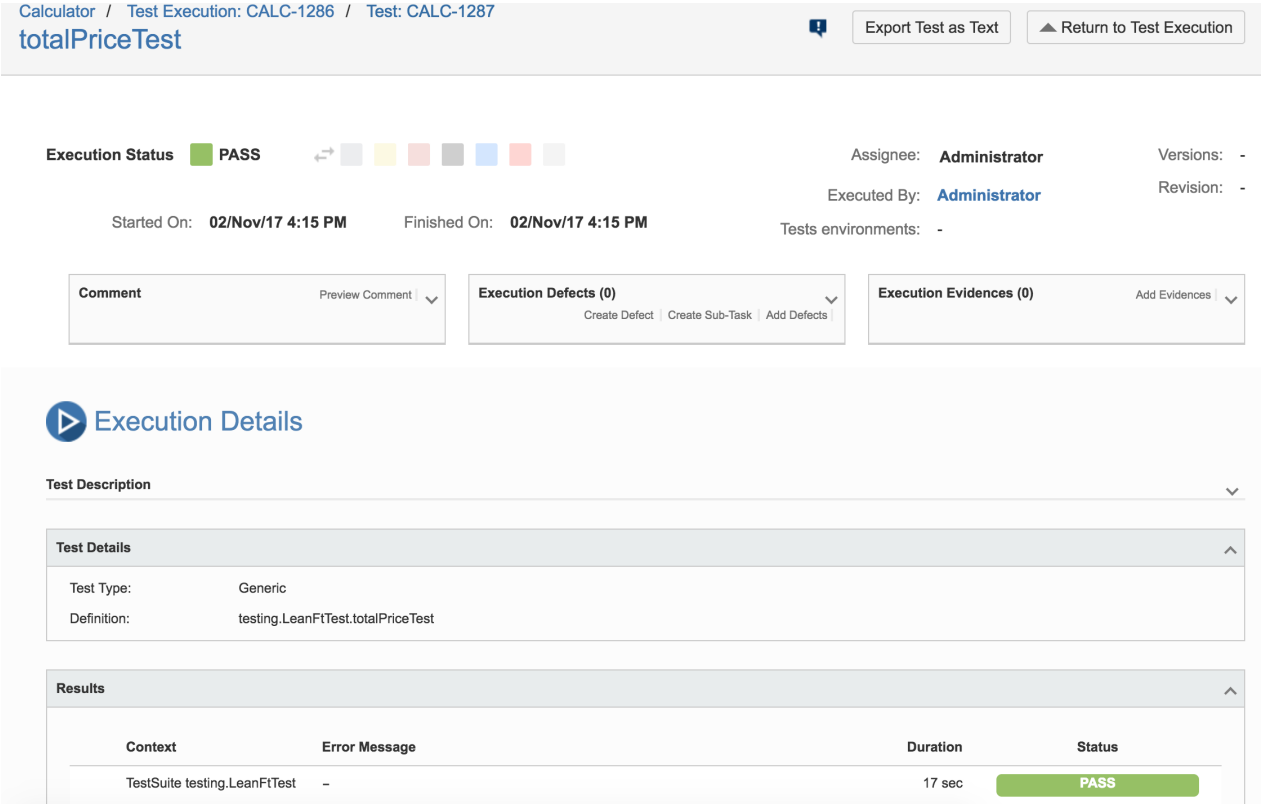
```
    @Before
    public void setUp() throws Exception {
    }

    @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 successfuly the Test Case and generating the JUnit XML report (e.g., TEST-testing.LeanFtTest-success.xml), it can be imported to Xray (either by the REST API or through the **Import Execution Results** action within the Test Execution).

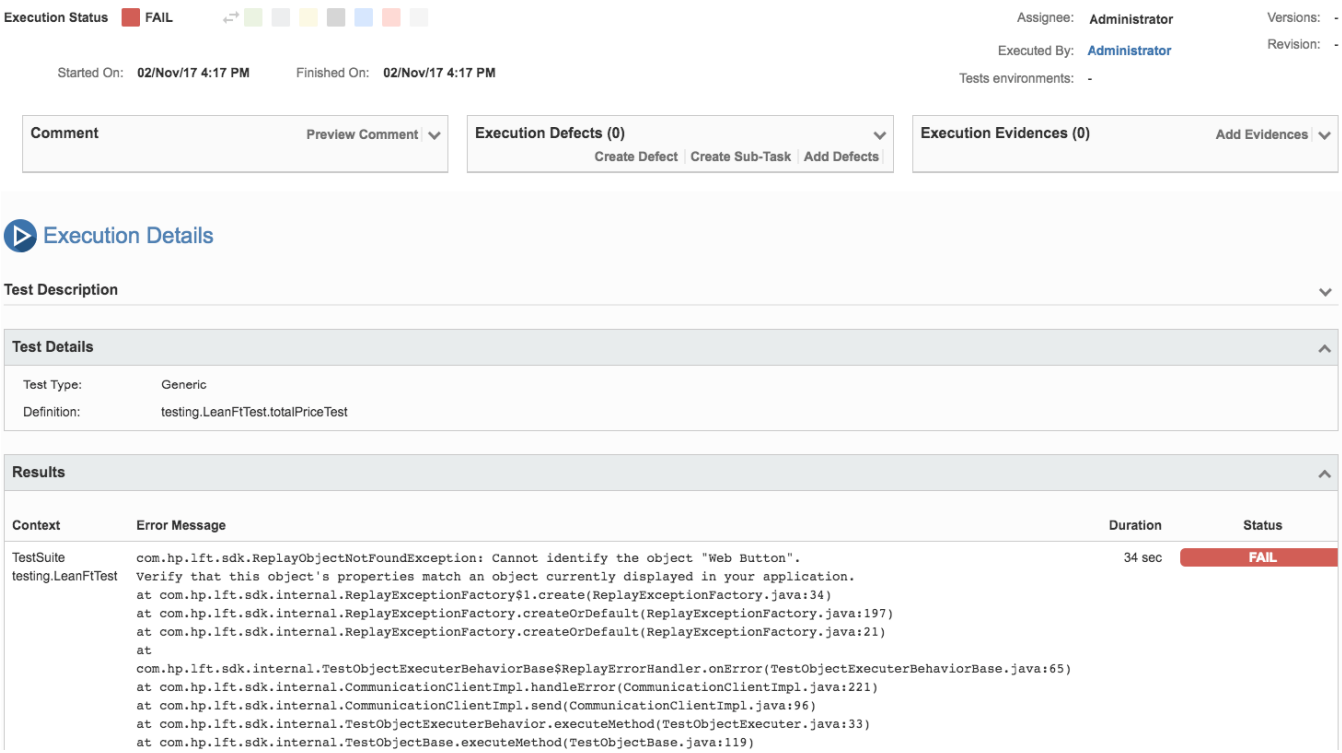


JUnit's Test Case is mapped to a Generic Test in Jira, and the **Generic Test Definition** field contains the package name, followed by the name of the class and the method name that implements the Test Case.

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.



If the Test fails, for example, due to a missing web element (e.g., TEST-testing.LeanFtTest-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.



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