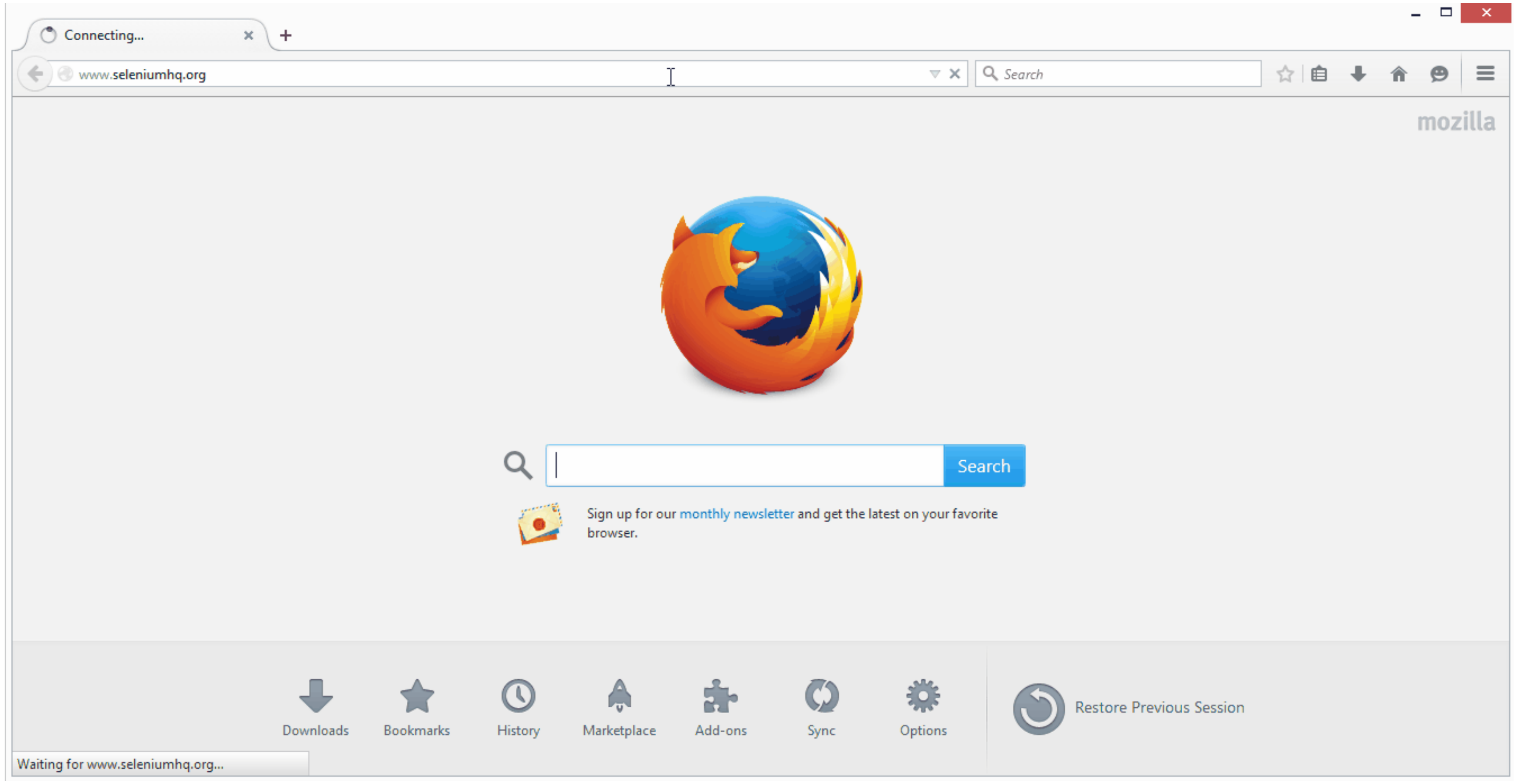
RAFSAN SAADI

San Francisco State Universion
CS Senior presentation
11/13/2017

GUI Automation with Selenium



WHAT is Test Automation ?

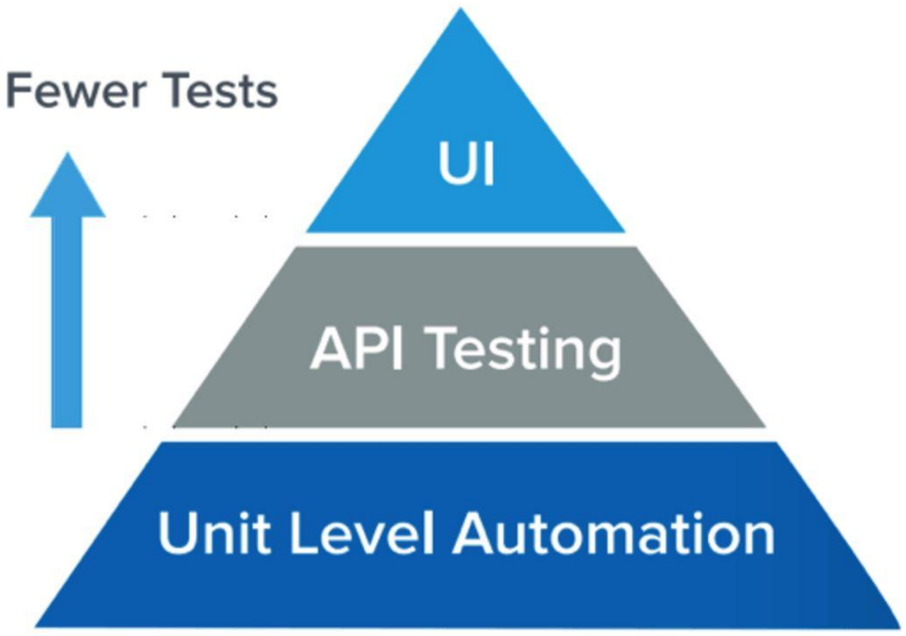
- A special software to automate repetitive but necessary testing tasks.
- Control the execution of tests in reliable fashion and produce consistent result.
- Perform testing that would be difficult to do manually.
- A critical part of CI process.

WHY Automation Testing?

- Automated Software Testing Saves Time and Money.
- Vastly Increases Test Coverage.
- Testing Improves Accuracy.
- Automation Does What Manual Testing Cannot.

Test Automation Pyramid

- **Unit Test** represents the base and biggest percentage of the test automation pyramid.
- **Automated API** or Service Layer testing.
- **GUI Automation** tests sit at the top of the pyramid and generally has the fewer number of tests.



Test Automation Pyramid

© Rafsan Saadi

GUI Automation with Selenium Framework

- Java
  - *Programming language for the Framework*
  - *https://docs.oracle.com/javase/tutorial/*
- Selenium WebDriver
  - *Frontend automation tool*
  - *http://www.seleniumhq.org/*
- Apache Maven
  - *Build automation tool used primarily for Java projects*
  - *https://maven.apache.org/*
- TestNG
  - *Testing framework inspired from JUnit that is more powerful and easier to use*
  - *http://testng.org/*
- Log4j
  - *Apache Log4j is a Java-based logging utility*
  - *https://logging.apache.org/log4j/2.x/*

Sample Test Definition

Test Name: Verify buyer can contact media seller.

```
// ==== CONFIGURATION ====
        // The item will be searched “Banana Gun”
        // The message will be sent “Test Msg from Buyer on 11/07/2017”
// ==== ACTION ====
        // Navigate to user LOGIN page
        // Login as the BUYER
        // Navigate to the App HOME page
        // Searched for the item that was specified at the top: "Banana Gun“
        // Send message to the seller with specified message body at the top
// ==== VERIFICATION ====
        // Verify the message is sent successfully “Message Send Success”
```

Sample Test Implementation

Test Name: Verify buyer can contact media seller.

```
   @Test(enabled = true, description = "Verify buyer can
   contact media seller")
   public void  testBuyerContactMediaOwnerSuccess() throws
   Exception{

// ==== CONFIGURATION ====
    // The item will be searched
    String mediaTitle = "Banana Gun";
    // The message will be sent
    String messageToSeller = "Test Msg from Buyer on " +
       GeneralUtils.currentTimeStamp();
```

Sample Test Implementation (Cont…)

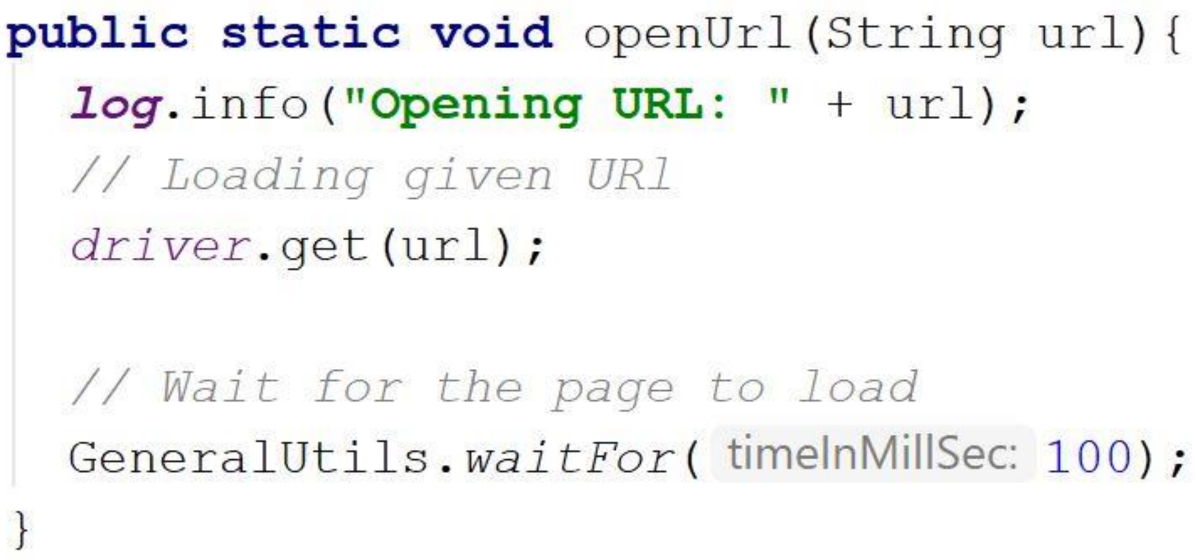
```
// ==== ACTION ====
    // Navigate to user LOGIN page
    UiActions.openUrl(USER_LOGIN_URL);
    // Login as the BUYER
    UiActions.loggingInAs(BUYER_USER_NAME,BUYER_USER_PASSWORD );
    // Navigate to the App HOME page
    UiActions.openingHomePage();
    // Searched for the item that was specified at the top: "Banana Gun“
    UiActions.searchWithoutCategory(mediaTitle);
    // Send message to the seller with specified message body at the top
    UiActions.sendMessageToSellerForSearchedItem(messageToSeller);
```

Sample Test Implementation (Cont…)

```
// ==== VERIFICATION ====
        // Verify the message is sent successfully
        UiVerifications.verifyMessageSentSuccessfully();
```

Method Implementation Details

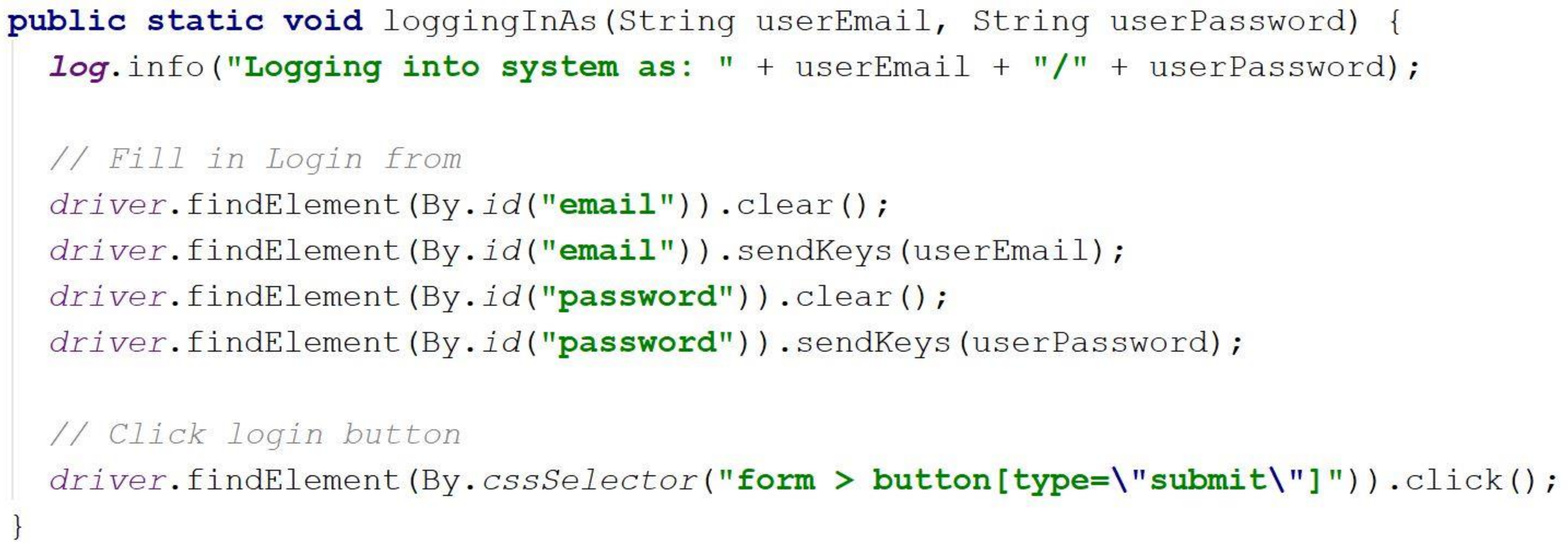
```
UiActions.openUrl(USER_LOGIN_URL);
```



```
public static void openUrl(String url){
    log.info("Opening URL: " + url);
    // Loading given URl
    driver.get(url);

    // Wait for the page to load
    GeneralUtils.waitFor( timeInMillSec: 100);
}
```

Method Implementation Details (Cont…)

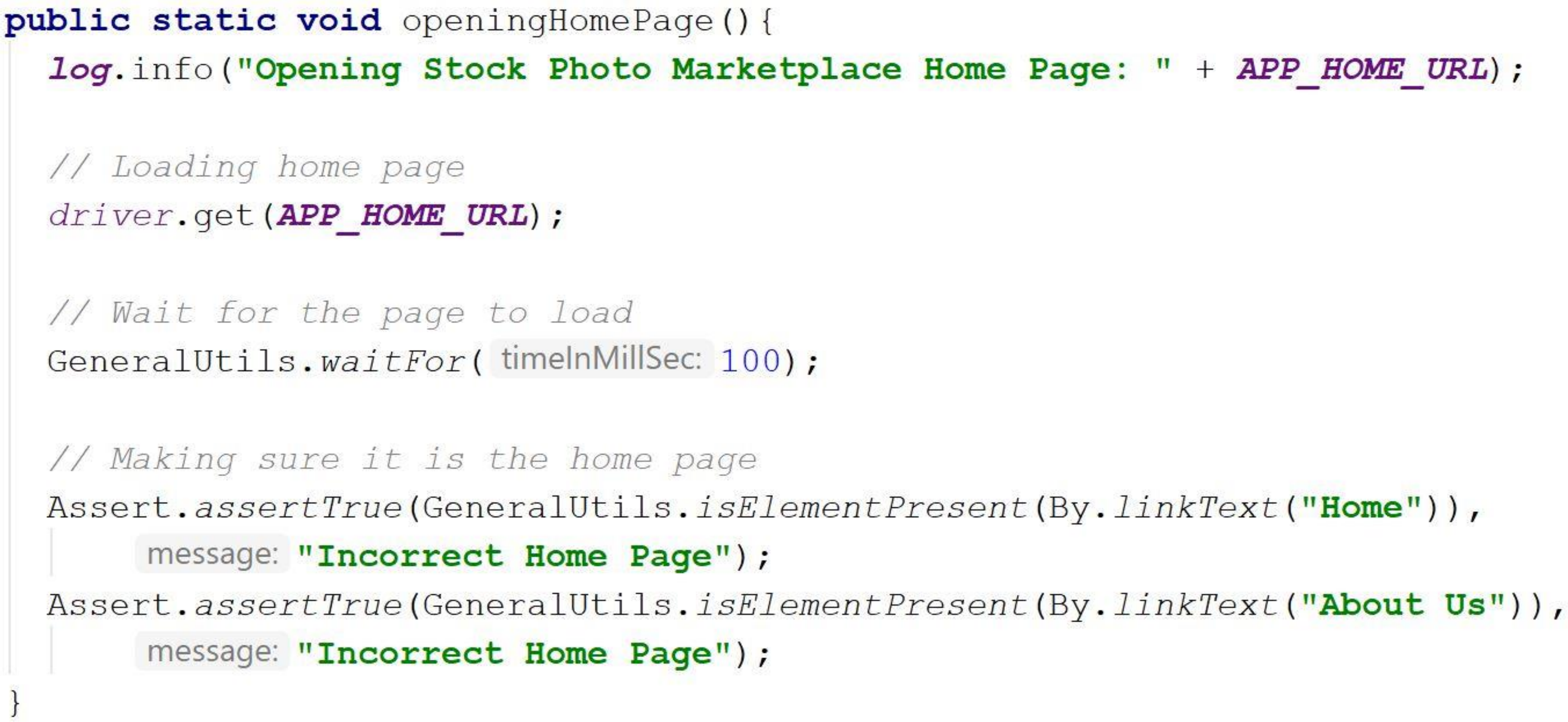
```
UiActions.loggingInAs(BUYER_USER_NAME,BUYER_USER_PASSWORD );
```



```
public static void loggingInAs(String userEmail, String userPassword) {
  log.info("Logging into system as: " + userEmail + "/" + userPassword);

  // Fill in Login from
  driver.findElement(By.id("email")).clear();
  driver.findElement(By.id("email")).sendKeys(userEmail);
  driver.findElement(By.id("password")).clear();
  driver.findElement(By.id("password")).sendKeys(userPassword);

  // Click login button
  driver.findElement(By.cssSelector("form > button[type=\"submit\"]")).click();
}
```

Method Implementation Details (Cont…)

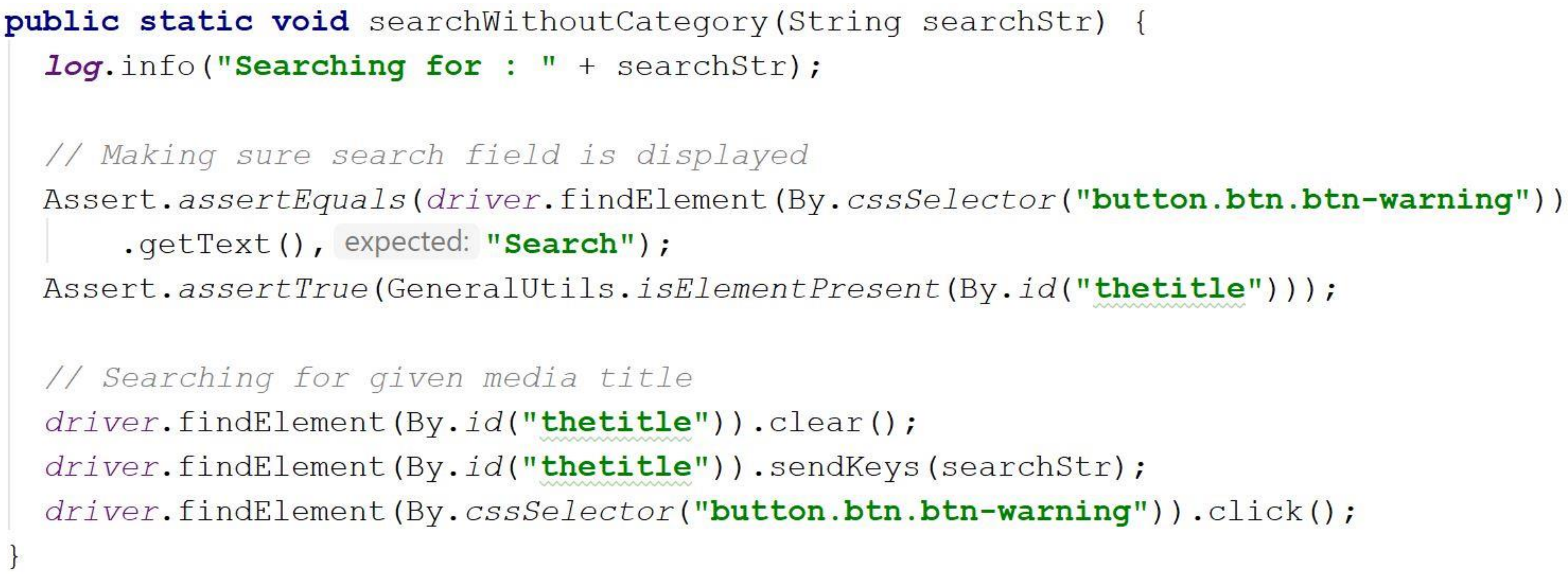
```
UiActions.openingHomePage();
```



```
public static void openingHomePage(){
    log.info("Opening Stock Photo Marketplace Home Page: " + APP_HOME_URL);

    // Loading home page
    driver.get(APP_HOME_URL);

    // Wait for the page to load
    GeneralUtils.waitFor( timeInMillSec: 100);

    // Making sure it is the home page
    Assert.assertTrue(GeneralUtils.isElementPresent(By.linkText("Home")),
            message: "Incorrect Home Page");
    Assert.assertTrue(GeneralUtils.isElementPresent(By.linkText("About Us")),
            message: "Incorrect Home Page");
}
```

Method Implementation Details (Cont…)

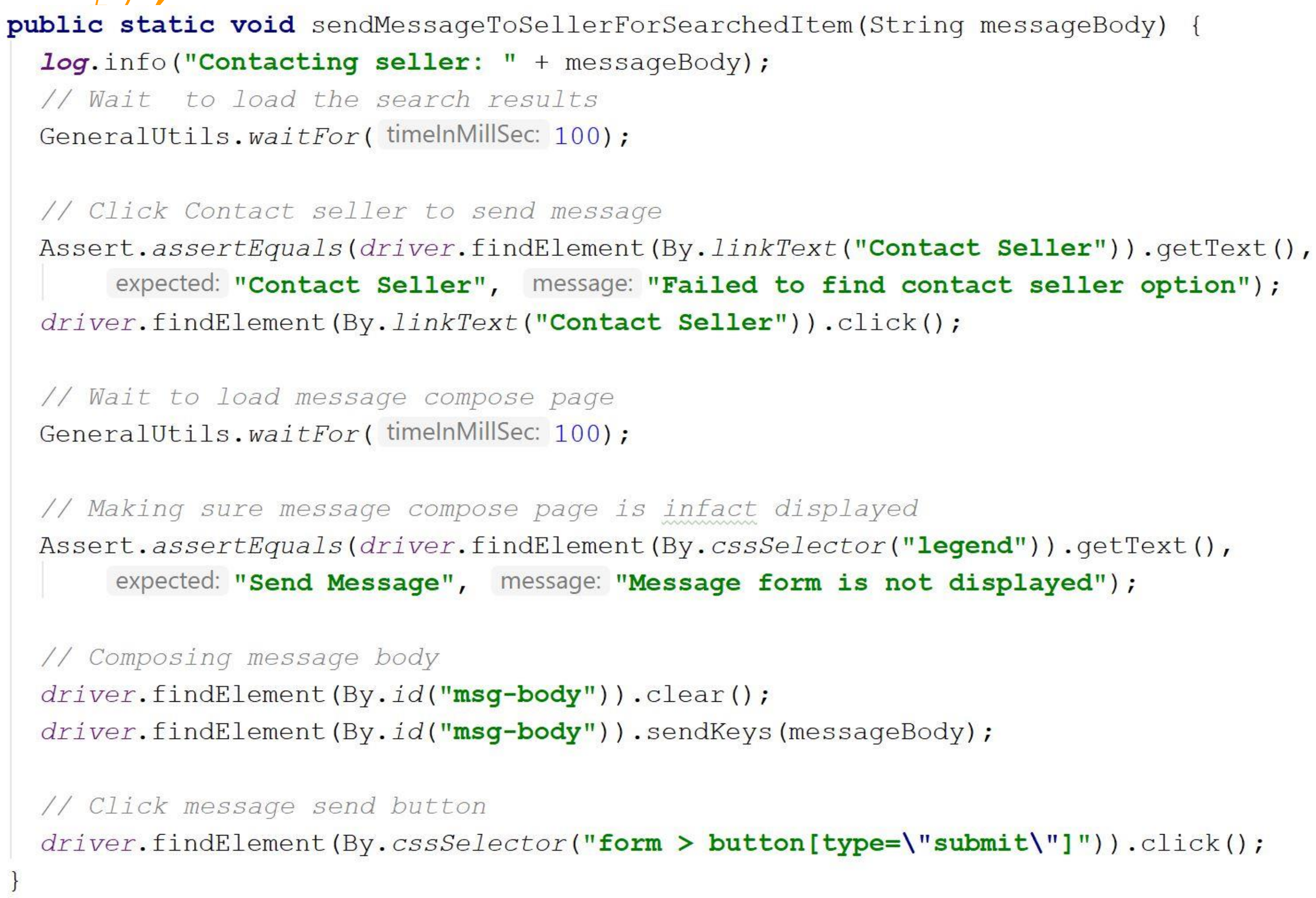
```
UiActions.searchWithoutCategory(mediaTitle);
```



```
public static void searchWithoutCategory(String searchStr) {
    log.info("Searching for : " + searchStr);

    // Making sure search field is displayed
    Assert.assertEquals(driver.findElement(By.cssSelector("button.btn.btn-warning"))
            .getText(), expected: "Search");
    Assert.assertTrue(GeneralUtils.isElementPresent(By.id("thetitle")));

    // Searching for given media title
    driver.findElement(By.id("thetitle")).clear();
    driver.findElement(By.id("thetitle")).sendKeys(searchStr);
    driver.findElement(By.cssSelector("button.btn.btn-warning")).click();
}
```

Method Implementation Details (Cont…)

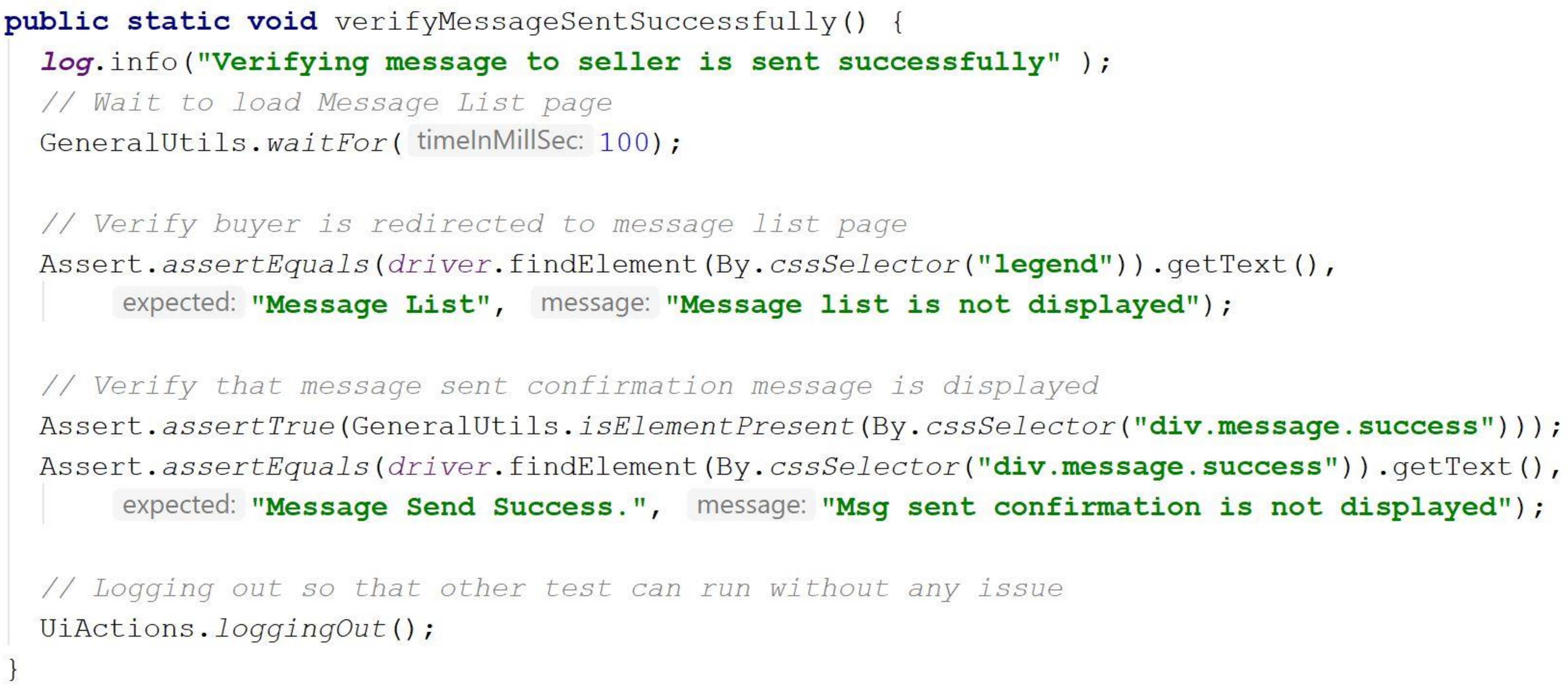
UiActions.sendMessageToSellerForSearchedItem(messageToSeller);



```
public static void sendMessageToSellerForSearchedItem(String messageBody) {
    log.info("Contacting seller: " + messageBody);
    // Wait  to load the search results
    GeneralUtils.waitFor( timeInMillSec: 100);

    // Click Contact seller to send message
    Assert.assertEquals(driver.findElement(By.linkText("Contact Seller")).getText(),
            expected: "Contact Seller",  message: "Failed to find contact seller option");
    driver.findElement(By.linkText("Contact Seller")).click();

    // Wait to load message compose page
    GeneralUtils.waitFor( timeInMillSec: 100);

    // Making sure message compose page is infact displayed
    Assert.assertEquals(driver.findElement(By.cssSelector("legend")).getText(),
            expected: "Send Message",  message: "Message form is not displayed");

    // Composing message body
    driver.findElement(By.id("msg-body")).clear();
    driver.findElement(By.id("msg-body")).sendKeys(messageBody);

    // Click message send button
    driver.findElement(By.cssSelector("form > button[type=\"submit\"]")).click();
}
```

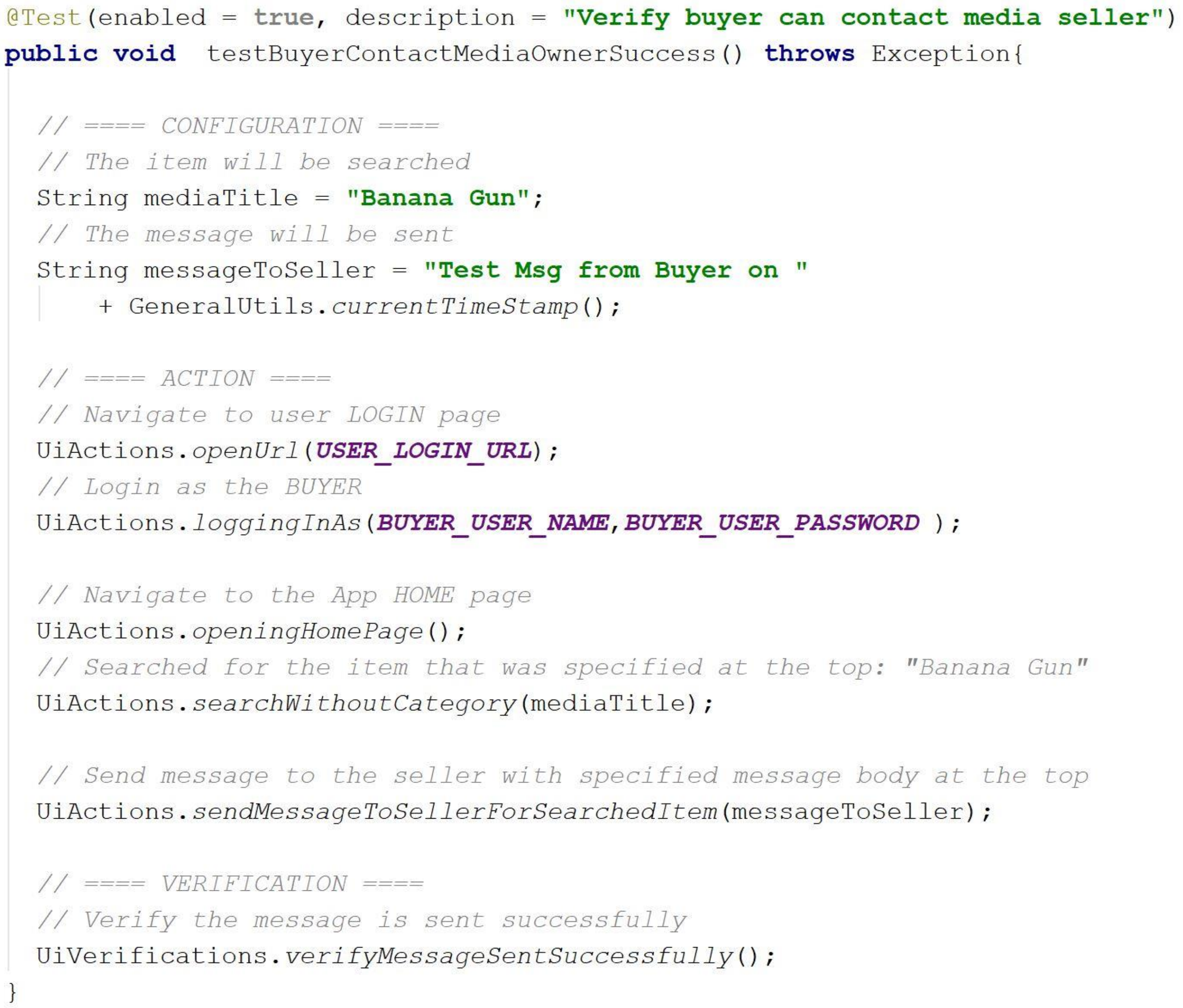
Method Implementation Details (Cont…)

```
UiVerifications.verifyMessageSentSuccessfully();
```



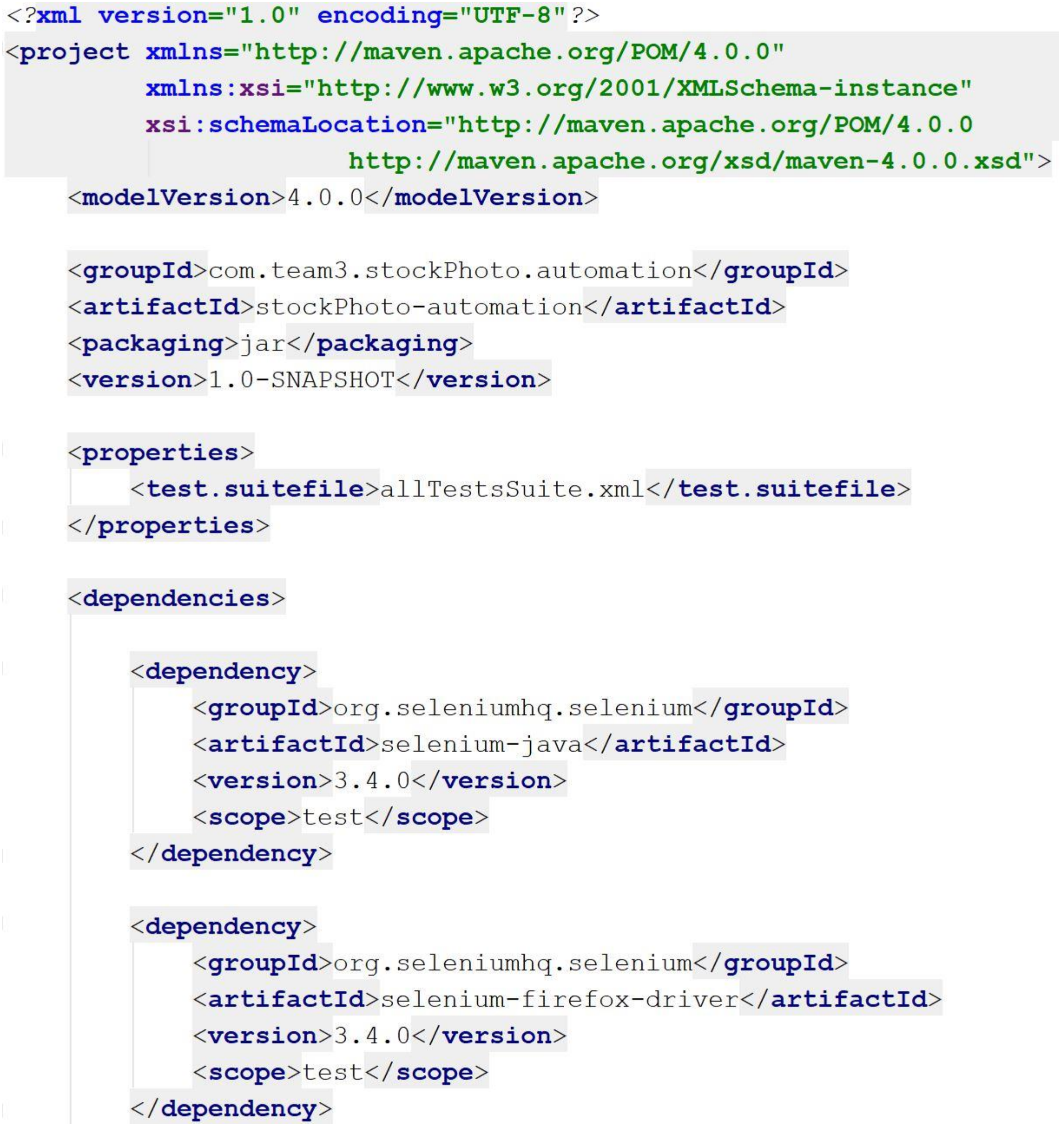
```
public static void verifyMessageSentSuccessfully() {
    log.info("Verifying message to seller is sent successfully" );
    // Wait to load Message List page
    GeneralUtils.waitFor( timeInMillSec: 100);

    // Verify buyer is redirected to message list page
    Assert.assertEquals(driver.findElement(By.cssSelector("legend")).getText(),
            expected: "Message List", message: "Message list is not displayed");

    // Verify that message sent confirmation message is displayed
    Assert.assertTrue(GeneralUtils.isElementPresent(By.cssSelector("div.message.success")));
    Assert.assertEquals(driver.findElement(By.cssSelector("div.message.success")).getText(),
            expected: "Message Send Success.", message: "Msg sent confirmation is not displayed");

    // Logging out so that other test can run without any issue
    UiActions.loggingOut();
}
```

A Complete Test



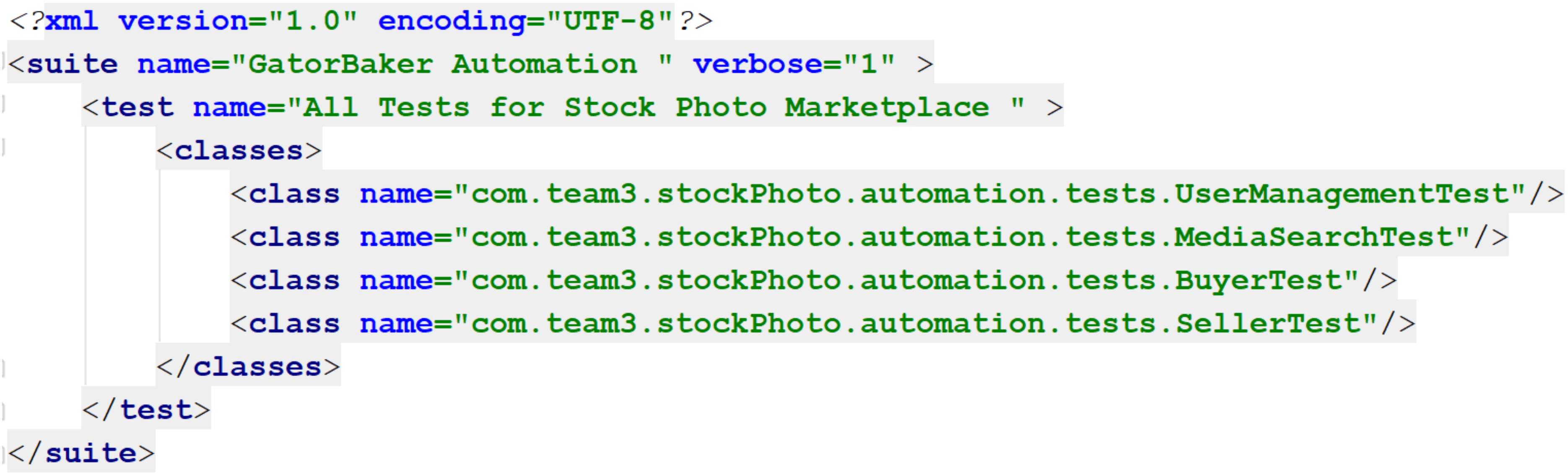
```
@Test(enabled = true, description = "Verify buyer can contact media seller")
public void  testBuyerContactMediaOwnerSuccess() throws Exception{

  // ==== CONFIGURATION ====
  // The item will be searched
  String mediaTitle = "Banana Gun";
  // The message will be sent
  String messageToSeller = "Test Msg from Buyer on "
      + GeneralUtils.currentTimeStamp();

  // ==== ACTION ====
  // Navigate to user LOGIN page
  UiActions.openUrl(USER_LOGIN_URL);
  // Login as the BUYER
  UiActions.loggingInAs(BUYER_USER_NAME,BUYER_USER_PASSWORD );

  // Navigate to the App HOME page
  UiActions.openingHomePage();
  // Searched for the item that was specified at the top: "Banana Gun"
  UiActions.searchWithoutCategory(mediaTitle);

  // Send message to the seller with specified message body at the top
  UiActions.sendMessageToSellerForSearchedItem(messageToSeller);

  // ==== VERIFICATION ====
  // Verify the message is sent successfully
  UiVerifications.verifyMessageSentSuccessfully();
}
```

Sample Pom.xml File



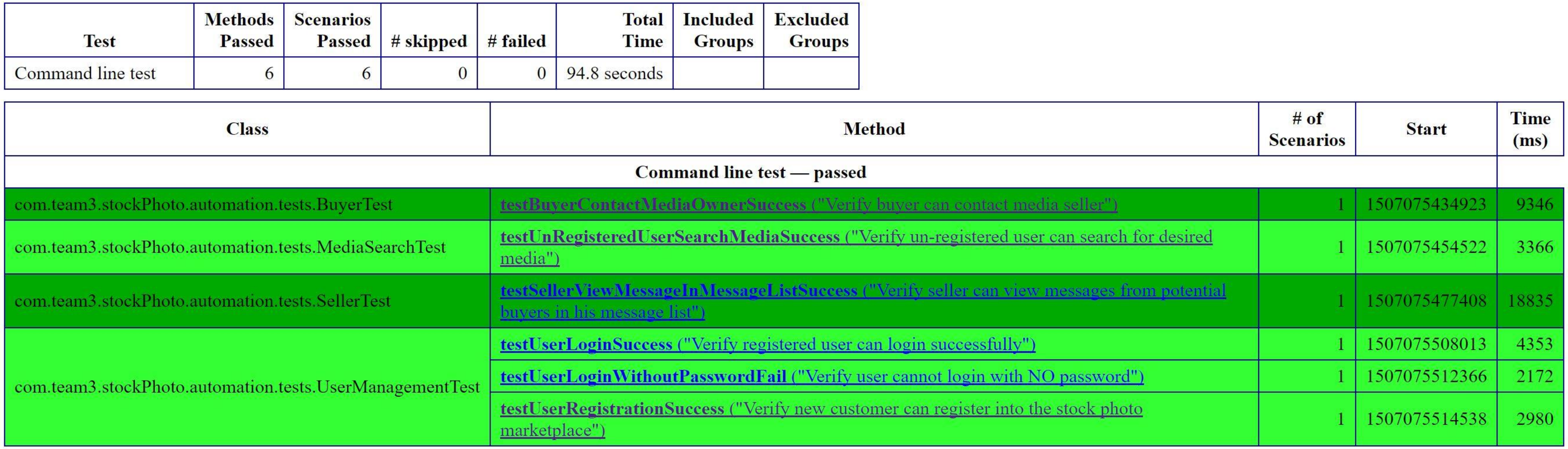
```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
                      http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>com.team3.stockPhoto.automation</groupId>
    <artifactId>stockPhoto-automation</artifactId>
    <packaging>jar</packaging>
    <version>1.0-SNAPSHOT</version>

    <properties>
        <test.suitefile>allTestsSuite.xml</test.suitefile>
    </properties>

    <dependencies>

        <dependency>
            <groupId>org.seleniumhq.selenium</groupId>
            <artifactId>selenium-java</artifactId>
            <version>3.4.0</version>
            <scope>test</scope>
        </dependency>

        <dependency>
            <groupId>org.seleniumhq.selenium</groupId>
            <artifactId>selenium-firefox-driver</artifactId>
            <version>3.4.0</version>
            <scope>test</scope>
        </dependency>
```

Sample TestsSuite.xml File



```xml
<?xml version="1.0" encoding="UTF-8"?>
<suite name="GatorBaker Automation " verbose="1" >
    <test name="All Tests for Stock Photo Marketplace " >
        <classes>
            <class name="com.team3.stockPhoto.automation.tests.UserManagementTest"/>
            <class name="com.team3.stockPhoto.automation.tests.MediaSearchTest"/>
            <class name="com.team3.stockPhoto.automation.tests.BuyerTest"/>
            <class name="com.team3.stockPhoto.automation.tests.SellerTest"/>
        </classes>
    </test>
</suite>
```

Sample Test Suite Execution Report



Test	Methods Passed	Scenarios Passed	# skipped	# failed	Total Time	Included Groups	Excluded Groups
Command line test	6	6	0	0	94.8 seconds		

Class	Method	# of Scenarios	Start	Time (ms)
Command line test — passed				
com.team3.stockPhoto.automation.tests.BuyerTest	**testBuyerContactMediaOwnerSuccess** ("Verify buyer can contact media seller")	1	1507075434923	9346
com.team3.stockPhoto.automation.tests.MediaSearchTest	**testUnRegisteredUserSearchMediaSuccess** ("Verify un-registered user can search for desired media")	1	1507075454522	3366
com.team3.stockPhoto.automation.tests.SellerTest	**testSellerViewMessageInMessageListSuccess** ("Verify seller can view messages from potential buyers in his message list")	1	1507075477408	18835
com.team3.stockPhoto.automation.tests.UserManagementTest	**testUserLoginSuccess** ("Verify registered user can login successfully")	1	1507075508013	4353
	testUserLoginWithoutPasswordFail ("Verify user cannot login with NO password")	1	1507075512366	2172
	testUserRegistrationSuccess ("Verify new customer can register into the stock photo marketplace")	1	1507075514538	2980

Above report is generated by the Selenium Surefire plugin

More Helpful Links

- Selenium with Java (APIs)
  - https://seleniumhq.github.io/selenium/docs/api/java/index.html
- Firebug
  - https://getfirebug.com/
- Microsoft Best Practices for Coded UI Tests
  - https://docs.microsoft.com/en-us/visualstudio/test/best-practices-for-coded-ui-tests

ENJOY THE DEMO

Thank you !