PRODUCTS | 10 MIN READ

Katalon Studio versus Selenium-based open source frameworks

Katalon Studio is a test automation solution that leverages Selenium's core engine. Although it uses several Selenium's functionality, it is not simply a Selenium wrapper. This document offers a detailed comparison of Katalon and Selenium-based open-source frameworks, focusing on key features such as target users, deployment, test management and scripting language support, performance, usability, and integration with other tools.

1. Target Users

Software test automation usually needs technical skills and some knowledge about the application under test (AUT) for its successful application. Knowledge about AUT's related technologies is not a must but recommended. For example, having knowledge of HTML and JavaScript should be better than not if you are testing on the web.

Katalon Studio's users could be the testers with limited technical knowledge. Katalon Studio hides all technical complexities behind the scene and provides friendly UI with the manual mode (user can drag-drop, select keywords and test objects to form test steps), but still keeps necessary weapons for more technically powerful users who are able to dig deeper into coding with the scripting mode that fully supports development conveniences like syntax highlighting, code suggestion and debugging.

Selenium and Appium are mainly automated testing libraries which are more suitable for testers who possess good programming skills. They will typically need to use an IDE like Eclipse or Visual Studio, to import Selenium & Appium libraries (and certainly their dependencies) into a development project, to install necessary unit testing frameworks like JUnit, NUnit and TestNG before starting to write a test case.

2. Deployment & Getting Started

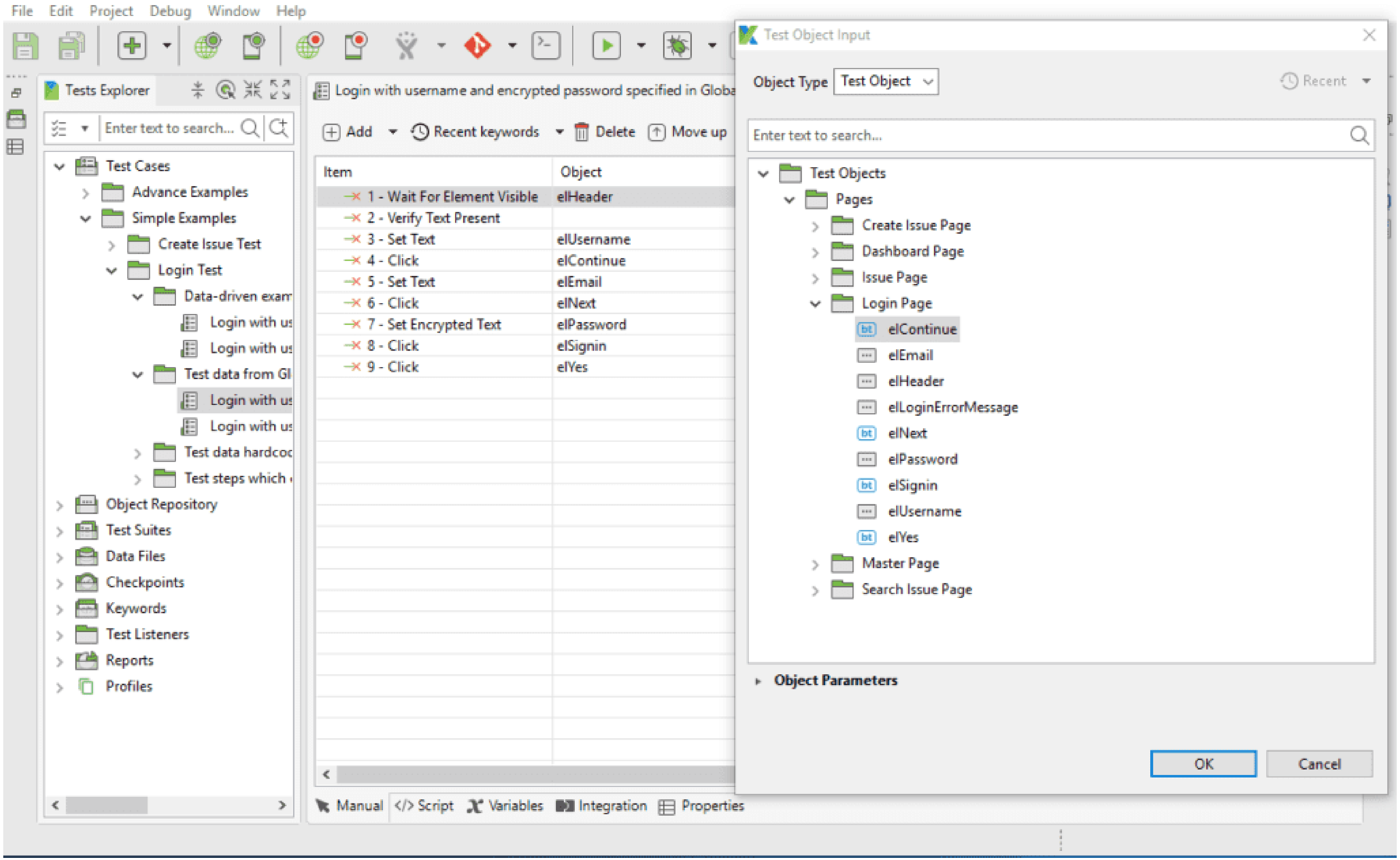
Most open source projects are hard to setup and deploy because they are contributed by many different developers and providers who submit their work into a central place, i.e., Github. You may have to build everything using the submitted source code. But even if frameworks are packaged into a ready-to-use stuff like Selenium and Appium, you still have to go through technical guidelines for their API and integrate them into with an IDE and a preferred programming language. Below are high-level steps required by Selenium, Appium and Katalon Studio to setup and run.

- Selenium:
    - Download and install Java SDK
    - Download and setup test libraries: Selenium client API of your preferred language, Selenium Server, or a standalone package (a bundle of client and server in only one library)
    - Download IEDriver and ChromeDriver, setup the environment & PATH variables
    - Download and setup the development IDE and your preferred testing frameworks like JUnit, NUnit and TestNG
    - Go through the Selenium API document in order to start to create a project and writing some code.
- Appium:
    - Download and install the SDK your programming language either Java, Python, Ruby or C#

    - Download and setup test libraries (Appium, Node.js, Android SDK, XCode if Mac and iOS), setup the environment & PATH variables
    - Download and setup the development IDE and your preferred testing frameworks like JUnit, NUnit and TestNG
    - Go through the Selenium API document in order to start to create a project and writing some code.
- Katalon Studio is built as a unified bundle which includes almost all necessary things like Java, Android SDK, Web drivers to drive browsers, and required dependencies. All you have to do is to download and install it on your computer. Minimum installation is required:
    - As Appium is frequently updated, Katalon Studio does not bundle Appium. If you are testing mobile apps, you have to install Appium, Node.js (on Windows) and XCode (on Mac)
    - You can quickly get familiar with Katalon Studio as it has friendly UI and sample projects. The detailed user documentation (docs.katalon.com) definitely helps you go through quickly from simple to advanced features.

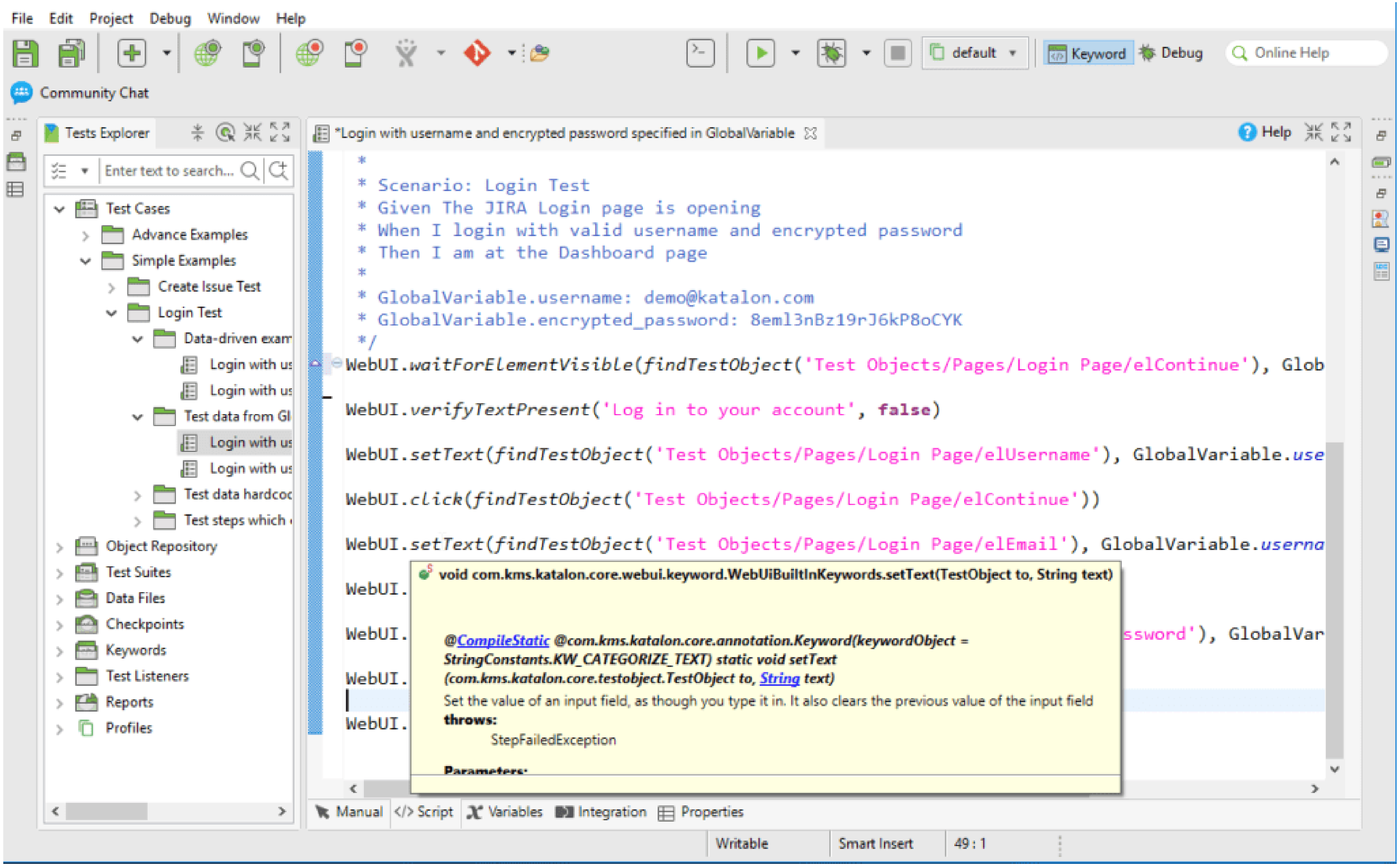
3. Test management and scripting language support

- Katalon Studio provides a rich graphical UI with menus, tree views, tables, etc. to manage test cases, objects and data files. Its composer for scripting with full syntax highlighting, code suggestion, debugging is ideal for testers who just want to drag-drop or have limited programming skills.



Item	Object
1 - Wait For Element Visible	elHeader
2 - Verify Text Present	
3 - Set Text	elUsername
4 - Click	elContinue
5 - Set Text	elEmail
6 - Click	elNext
7 - Set Encrypted Text	elPassword
8 - Click	elSignin
9 - Click	elYes

(Katalon Studio manual mode)



```
 *
 * Scenario: Login Test
 * Given The JIRA Login page is opening
 * When I login with valid username and encrypted password
 * Then I am at the Dashboard page
 *
 * GlobalVariable.username: demo@katalon.com
 * GlobalVariable.encrypted_password: 8eml3nBz19rJ6kP8oCYK
 */
WebUI.waitForElementVisible(findTestObject('Test Objects/Pages/Login Page/elContinue'), Glob

WebUI.verifyTextPresent('Log in to your account', false)

WebUI.setText(findTestObject('Test Objects/Pages/Login Page/elUsername'), GlobalVariable.use

WebUI.click(findTestObject('Test Objects/Pages/Login Page/elContinue'))

WebUI.setText(findTestObject('Test Objects/Pages/Login Page/elEmail'), GlobalVariable.userna

WebUI.

WebUI.                                                                ssword'), GlobalVar

WebUI.

WebUI.
```

void com.kms.katalon.core.webui.keyword.WebUiBuiltInKeywords.setText(TestObject to, String text)

@CompileStatic @com.kms.katalon.core.annotation.Keyword(keywordObject = StringConstants.KW_CATEGORIZE_TEXT) static void setText (com.kms.katalon.core.testobject.TestObject to, String text)

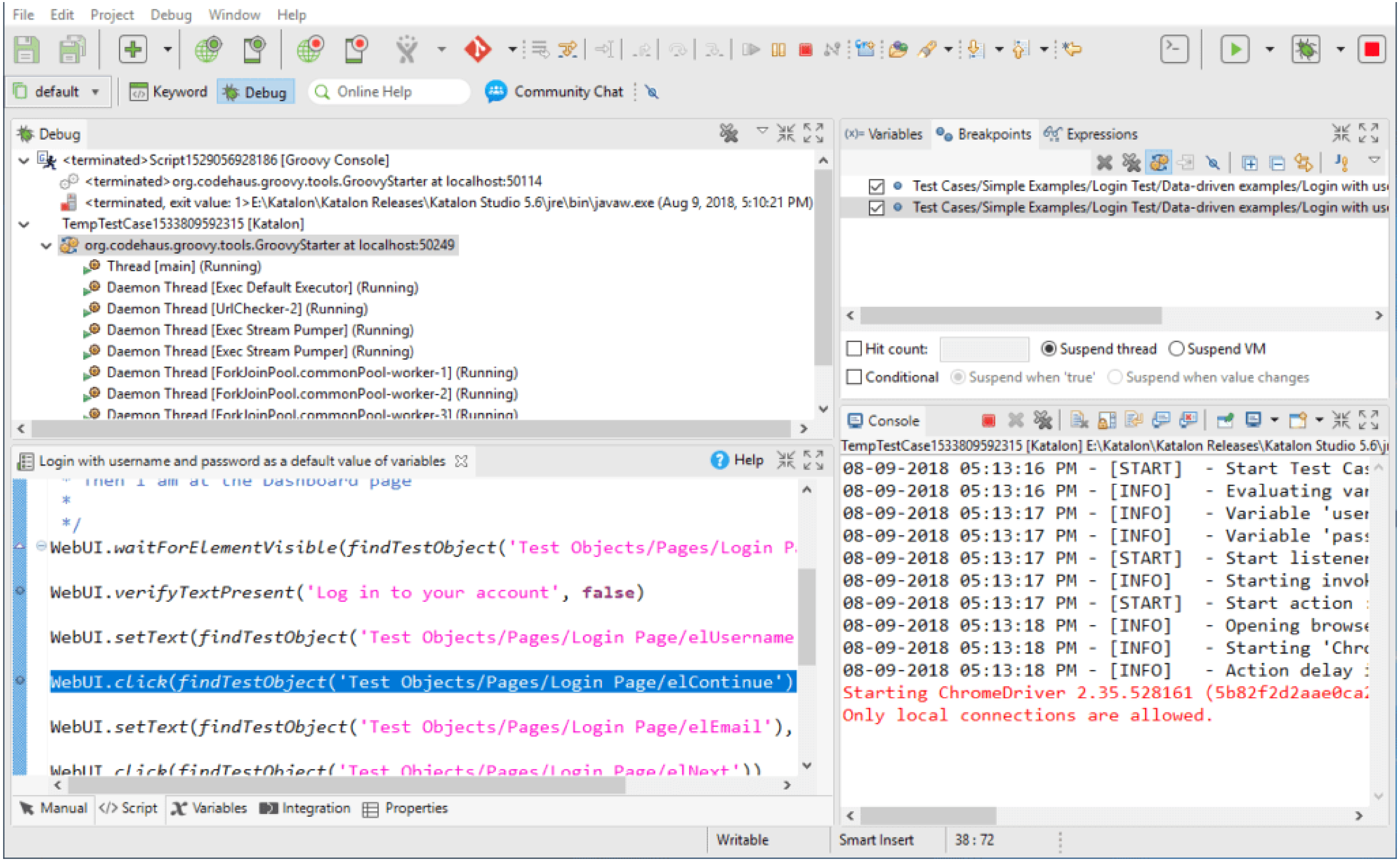
Set the value of an input field, as though you type it in. It also clears the previous value of the input field

throws:

StepFailedException

Parameters:

(Katalon Studio scripting mode)



```
<terminated>Script1529056928186 [Groovy Console]
    <terminated>org.codehaus.groovy.tools.GroovyStarter at localhost:50114
    <terminated, exit value: 1>E:\Katalon\Katalon Releases\Katalon Studio 5.6\jre\bin\javaw.exe (Aug 9, 2018, 5:10:21 PM)
TempTestCase1533809592315 [Katalon]
  org.codehaus.groovy.tools.GroovyStarter at localhost:50249
      Thread [main] (Running)
      Daemon Thread [Exec Default Executor] (Running)
      Daemon Thread [UrlChecker-2] (Running)
      Daemon Thread [Exec Stream Pumper] (Running)
      Daemon Thread [Exec Stream Pumper] (Running)
      Daemon Thread [ForkJoinPool.commonPool-worker-1] (Running)
      Daemon Thread [ForkJoinPool.commonPool-worker-2] (Running)
      Daemon Thread [ForkJoinPool.commonPool-worker-3] (Running)
```

☑ Test Cases/Simple Examples/Login Test/Data-driven examples/Login with us

☑ Test Cases/Simple Examples/Login Test/Data-driven examples/Login with us

□ Hit count: ● Suspend thread ○ Suspend VM

□ Conditional ○ Suspend when 'true' ○ Suspend when value changes

Login with username and password as a default value of variables

```
 * Then I am at the Dashboard page
 *
 */
WebUI.waitForElementVisible(findTestObject('Test Objects/Pages/Login P

WebUI.verifyTextPresent('Log in to your account', false)

WebUI.setText(findTestObject('Test Objects/Pages/Login Page/elUsername

WebUI.click(findTestObject('Test Objects/Pages/Login Page/elContinue')

WebUI.setText(findTestObject('Test Objects/Pages/Login Page/elEmail'),

WebUI.click(findTestObject('Test Objects/Pages/Login Page/elNext'))
```

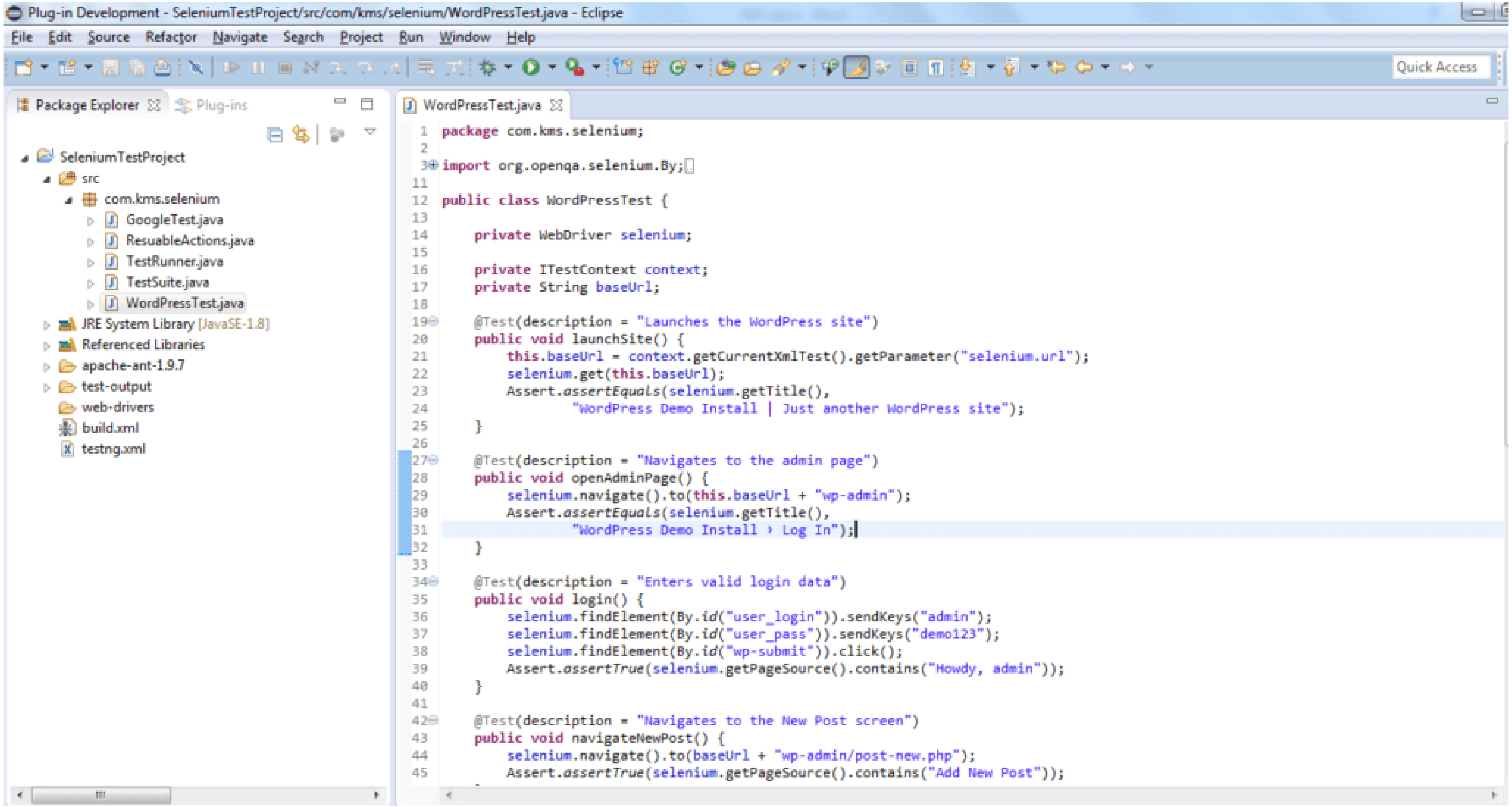
TempTestCase1533809592315 [Katalon] E:\Katalon\Katalon Releases\Katalon Studio 5.6\j

```
08-09-2018 05:13:16 PM - [START]  - Start Test Ca
08-09-2018 05:13:16 PM - [INFO]   - Evaluating var
08-09-2018 05:13:17 PM - [INFO]   - Variable 'user
08-09-2018 05:13:17 PM - [INFO]   - Variable 'pass
08-09-2018 05:13:17 PM - [START]  - Start listener
08-09-2018 05:13:17 PM - [INFO]   - Starting invok
08-09-2018 05:13:17 PM - [START]  - Start action :
08-09-2018 05:13:18 PM - [INFO]   - Opening browse
08-09-2018 05:13:18 PM - [INFO]   - Starting 'Chro
08-09-2018 05:13:18 PM - [INFO]   - Action delay i
Starting ChromeDriver 2.35.528161 (5b82f2d2aae0ca
Only local connections are allowed.
```

(Katalon Studio debug mode)

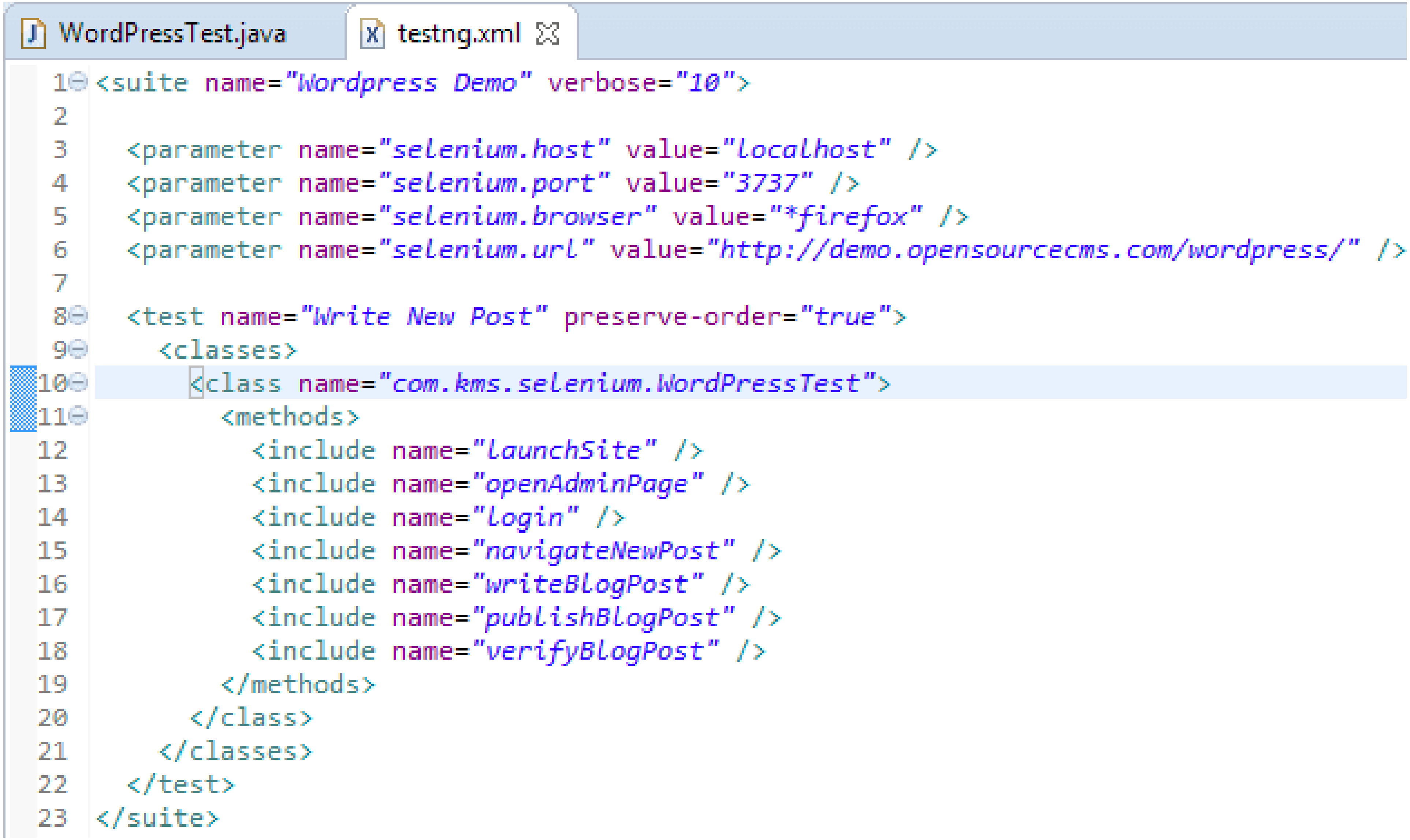
- Selenium and Appium, when integrating with a dedicated development IDE like Eclipse and Visual Studio, offer intuitive environments for developers but unfamiliar and unnecessarily complex features for manual or non-technical testers.

- Managing tests is challenging for non-technical testers when using Selenium and Appium.



```java
package com.kms.selenium;

import org.openqa.selenium.By;

public class WordPressTest {

    private WebDriver selenium;

    private ITestContext context;
    private String baseUrl;

    @Test(description = "Launches the WordPress site")
    public void launchSite() {
        this.baseUrl = context.getCurrentXmlTest().getParameter("selenium.url");
        selenium.get(this.baseUrl);
        Assert.assertEquals(selenium.getTitle(),
                "WordPress Demo Install | Just another WordPress site");
    }

    @Test(description = "Navigates to the admin page")
    public void openAdminPage() {
        selenium.navigate().to(this.baseUrl + "wp-admin");
        Assert.assertEquals(selenium.getTitle(),
                "WordPress Demo Install › Log In");
    }

    @Test(description = "Enters valid login data")
    public void login() {
        selenium.findElement(By.id("user_login")).sendKeys("admin");
        selenium.findElement(By.id("user_pass")).sendKeys("demo123");
        selenium.findElement(By.id("wp-submit")).click();
        Assert.assertTrue(selenium.getPageSource().contains("Howdy, admin"));
    }

    @Test(description = "Navigates to the New Post screen")
    public void navigateNewPost() {
        selenium.navigate().to(baseUrl + "wp-admin/post-new.php");
        Assert.assertTrue(selenium.getPageSource().contains("Add New Post"));
```

(The only development mode of Selenium + TestNG + Eclipse)



```xml
<suite name="Wordpress Demo" verbose="10">

  <parameter name="selenium.host" value="localhost" />
  <parameter name="selenium.port" value="3737" />
  <parameter name="selenium.browser" value="*firefox" />
  <parameter name="selenium.url" value="http://demo.opensourcecms.com/wordpress/" />

  <test name="Write New Post" preserve-order="true">
    <classes>
      <class name="com.kms.selenium.WordPressTest">
        <methods>
          <include name="launchSite" />
          <include name="openAdminPage" />
          <include name="login" />
          <include name="navigateNewPost" />
          <include name="writeBlogPost" />
          <include name="publishBlogPost" />
          <include name="verifyBlogPost" />
        </methods>
      </class>
    </classes>
  </test>
</suite>
```

(Test cases, Test suites, test data are mainly organized in XML file and programming code)

- Scripting language support:
  - Katalon Studio: Java/Groovy
  - Selenium, Appium + an IDE: Many choices for scripting languages

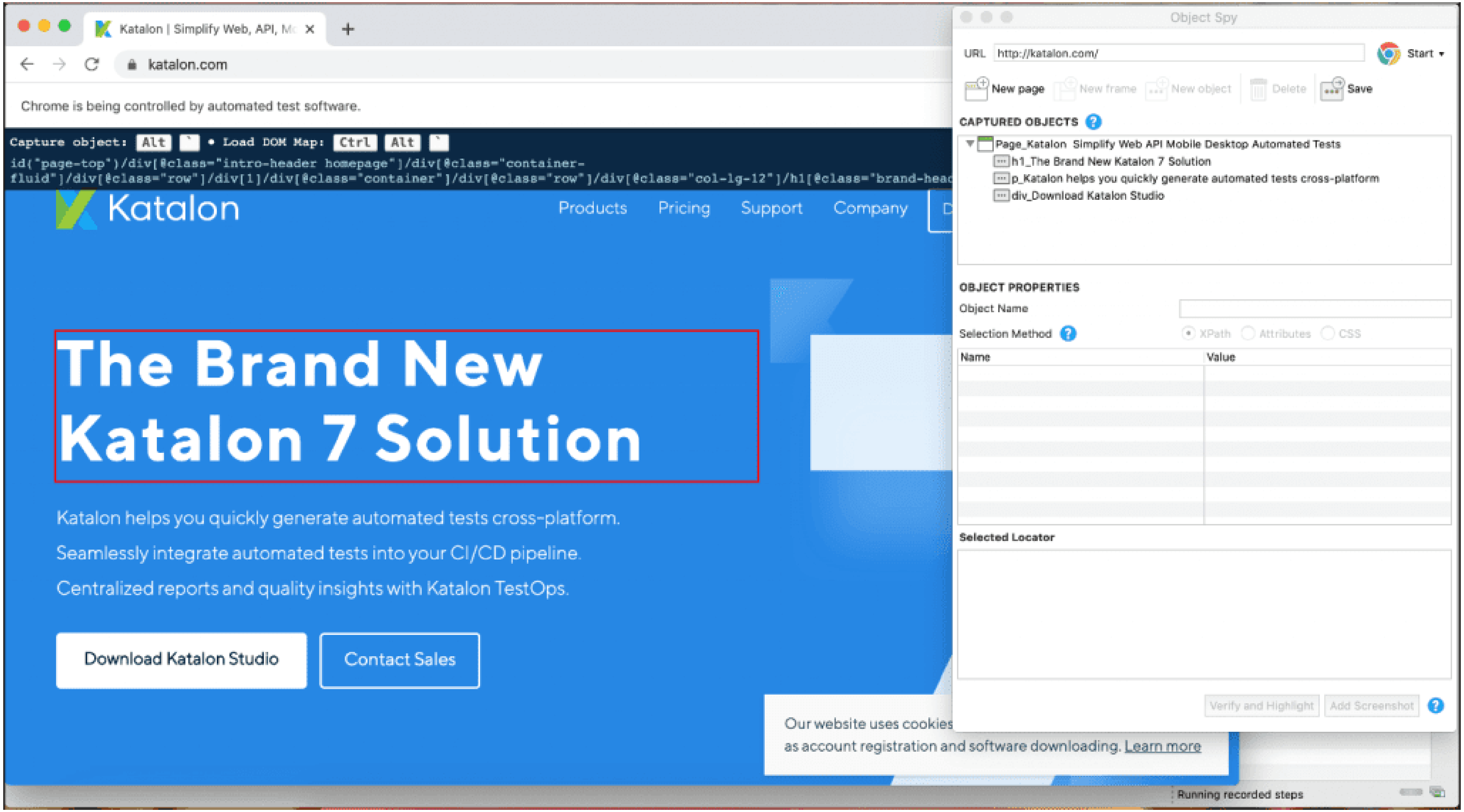
4. Performance

Katalon Studio uses Groovy, a language built on top of Java, and has to load many libraries for parsing test data, test objects, logging. So, it may be a bit slower as compared to Java for long test cases with hundreds of steps.

5. Usability and Functionality

5.1 Object spy: Spying test objects is the most important but time-consuming activity in test automation.

- Katalon Studio supports object spy for Web (by using browser plugins for IE, Firefox, Chrome) and for Mobile apps (by using the screenshot simulator). Katalon Studio also supports high-lighting captured objects in AUT:



Chrome is being controlled by automated test software.

```
Capture object: Alt ` • Load DOM Map: Ctrl Alt `
id("page-top")/div[@class="intro-header homepage"]/div[@class="container-
fluid"]/div[@class="row"]/div[1]/div[@class="container"]/div[@class="row"]/div[@class="col-lg-12"]/h1[@class="brand-head
```

The Brand New Katalon 7 Solution

Katalon helps you quickly generate automated tests cross-platform.

Seamlessly integrate automated tests into your CI/CD pipeline.

Centralized reports and quality insights with Katalon TestOps.

Download Katalon Studio | Contact Sales

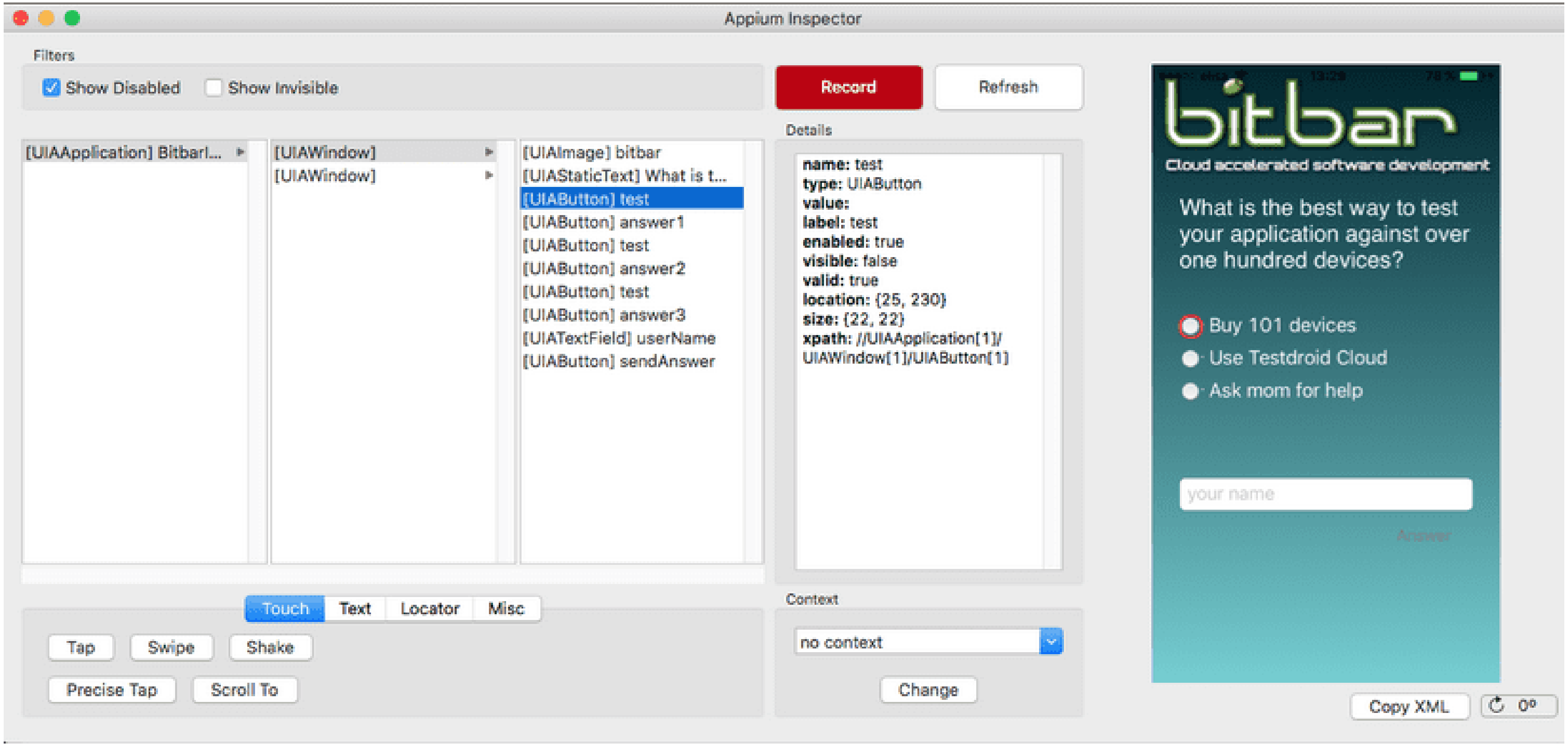
Our website uses cookies as account registration and software downloading. Learn more

URL http://katalon.com/

- Page_Katalon Simplify Web API Mobile Desktop Automated Tests
  - h1_The Brand New Katalon 7 Solution
  - p_Katalon helps you quickly generate automated tests cross-platform
  - div_Download Katalon Studio

(Katalon Object Spy with object high-lighting support)

- Selenium has a Firefox add-on called Selenium IDE for recording objects on Web applications.
- Appium provides a companion mobile object spy tool which is useful, but it has no high-lighting feature.



name: test
type: UIAButton
value:
label: test
enabled: true
visible: false
valid: true
location: {25, 230}
size: {22, 22}
xpath: //UIAApplication[1]/UIAWindow[1]/UIAButton[1]

What is the best way to test your application against over one hundred devices?

○ Buy 101 devices
○ Use Testdroid Cloud
○ Ask mom for help

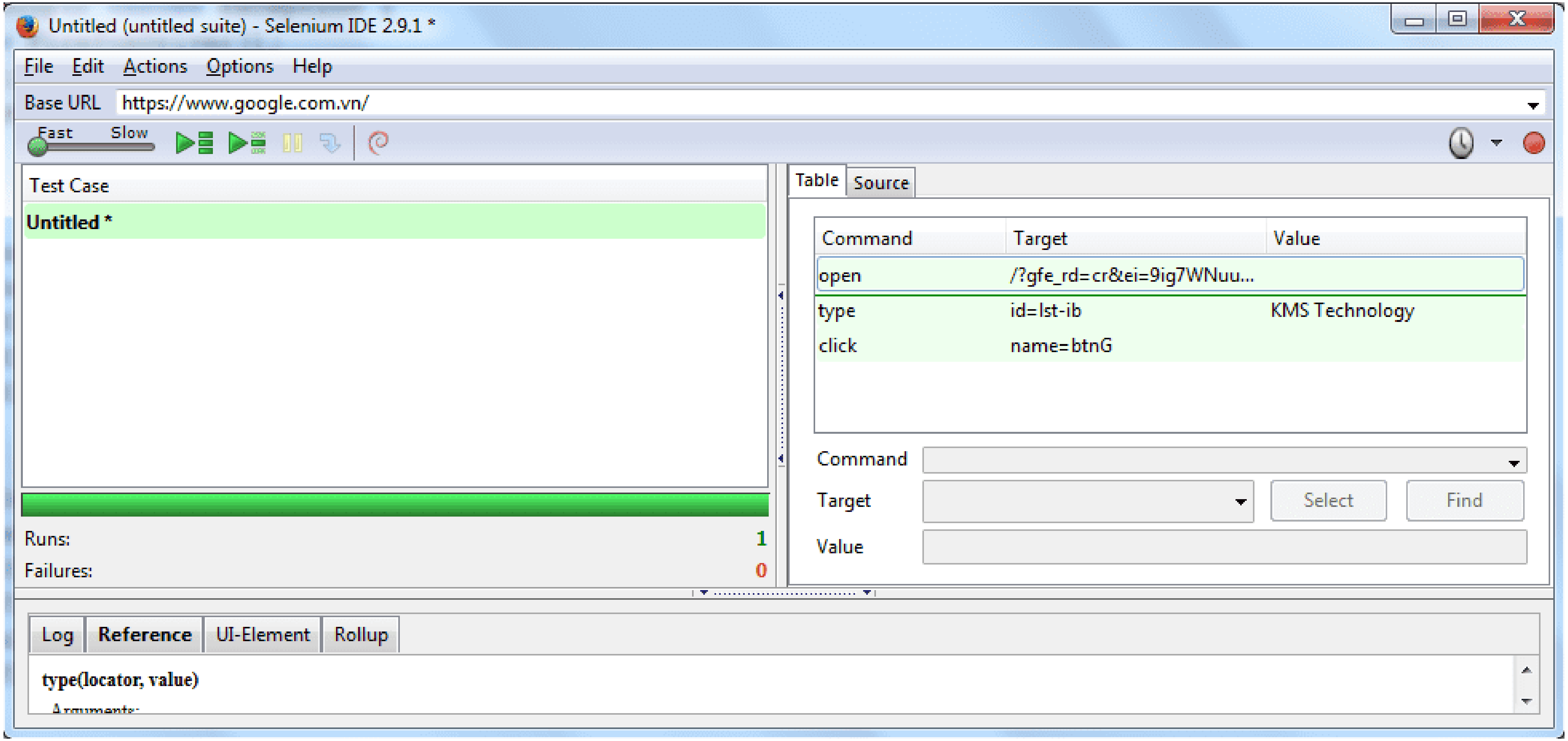
5.2 Test data management and data-driven

- Katalon Studio supports dynamic data sources via CSV, Excel, and databases or through Katalon Studio internal data.
- It also supports data-driven testing on a data file object. For example, when testing the login feature, you can specify 100 predefined accounts in an Excel spreadsheet or a database.
- In the case of Selenium and Appium, testers have to hard-code test data directly in the code. Any support for external data sources requires considerable programming effort.

5.3 Recording and test generation

Test recording is the best way to help users learn and quickly adapt to test automation. It helps an automation tester to do two primary jobs: capturing objects (during recording) and generating test steps with appropriate actions or keywords, saving at least 50-60% time of composing test cases.

- Katalon Studio has the capability to record and generate test scripts on both Web and mobile apps:



Base URL https://www.google.com.vn/

Command	Target	Value
open	/?gfe_rd=cr&ei=9ig7WNuu...	
type	id=lst-ib	KMS Technology
click	name=btnG	

Command

Target

Value

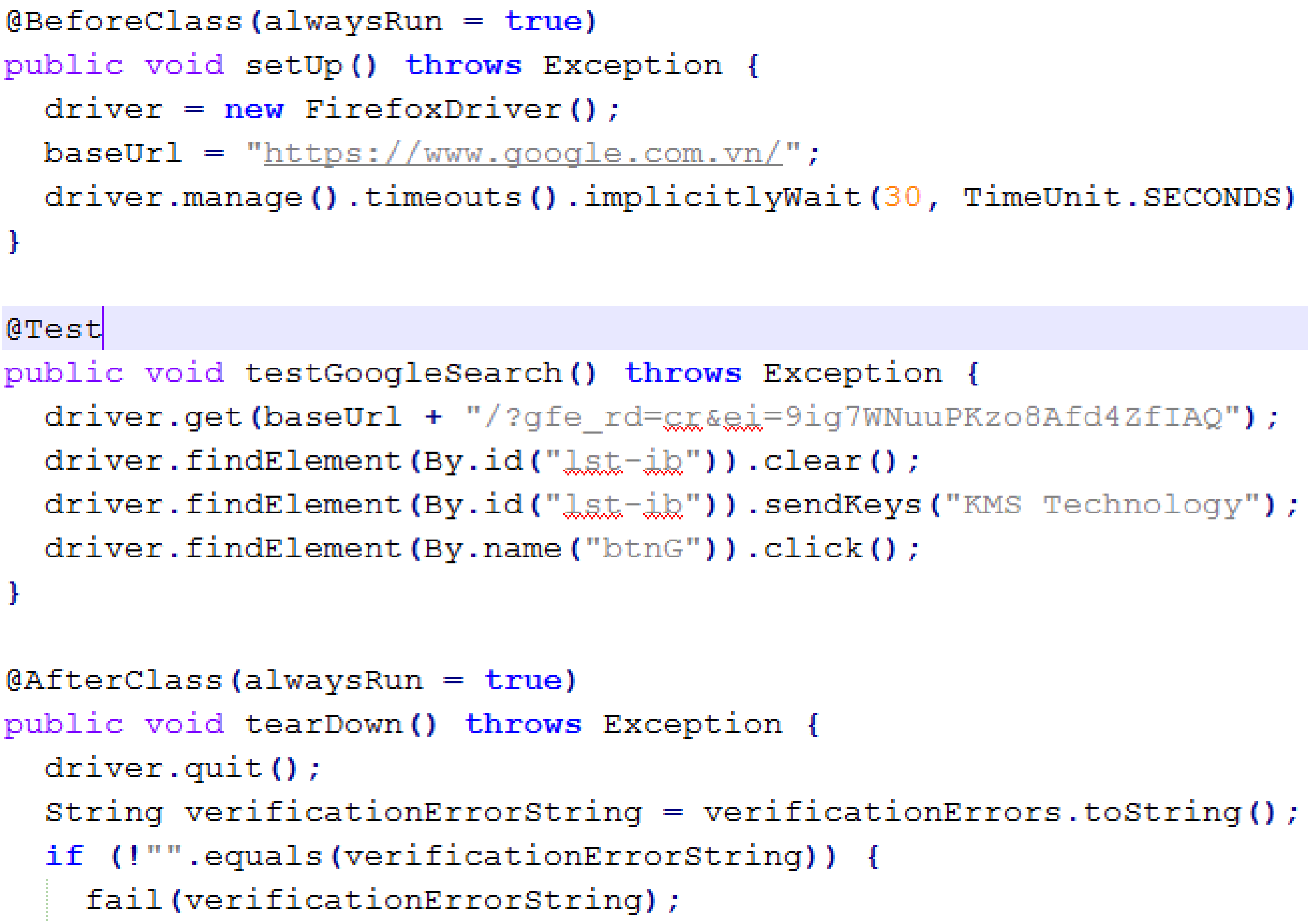
Runs: 1

Failures: 0

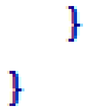
type(locator, value)

Arguments:

(Test Recording with Selenium IDE)



```
@BeforeClass(alwaysRun = true)
public void setUp() throws Exception {
  driver = new FirefoxDriver();
  baseUrl = "https://www.google.com.vn/";
  driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS)
}

@Test
public void testGoogleSearch() throws Exception {
  driver.get(baseUrl + "/?gfe_rd=cr&ei=9ig7WNuuPKzo8Afd4ZfIAQ");
  driver.findElement(By.id("lst-ib")).clear();
  driver.findElement(By.id("lst-ib")).sendKeys("KMS Technology");
  driver.findElement(By.name("btnG")).click();
}

@AfterClass(alwaysRun = true)
public void tearDown() throws Exception {
  driver.quit();
  String verificationErrorString = verificationErrors.toString();
  if (!"".equals(verificationErrorString)) {
    fail(verificationErrorString);
```



```
    }
}
```

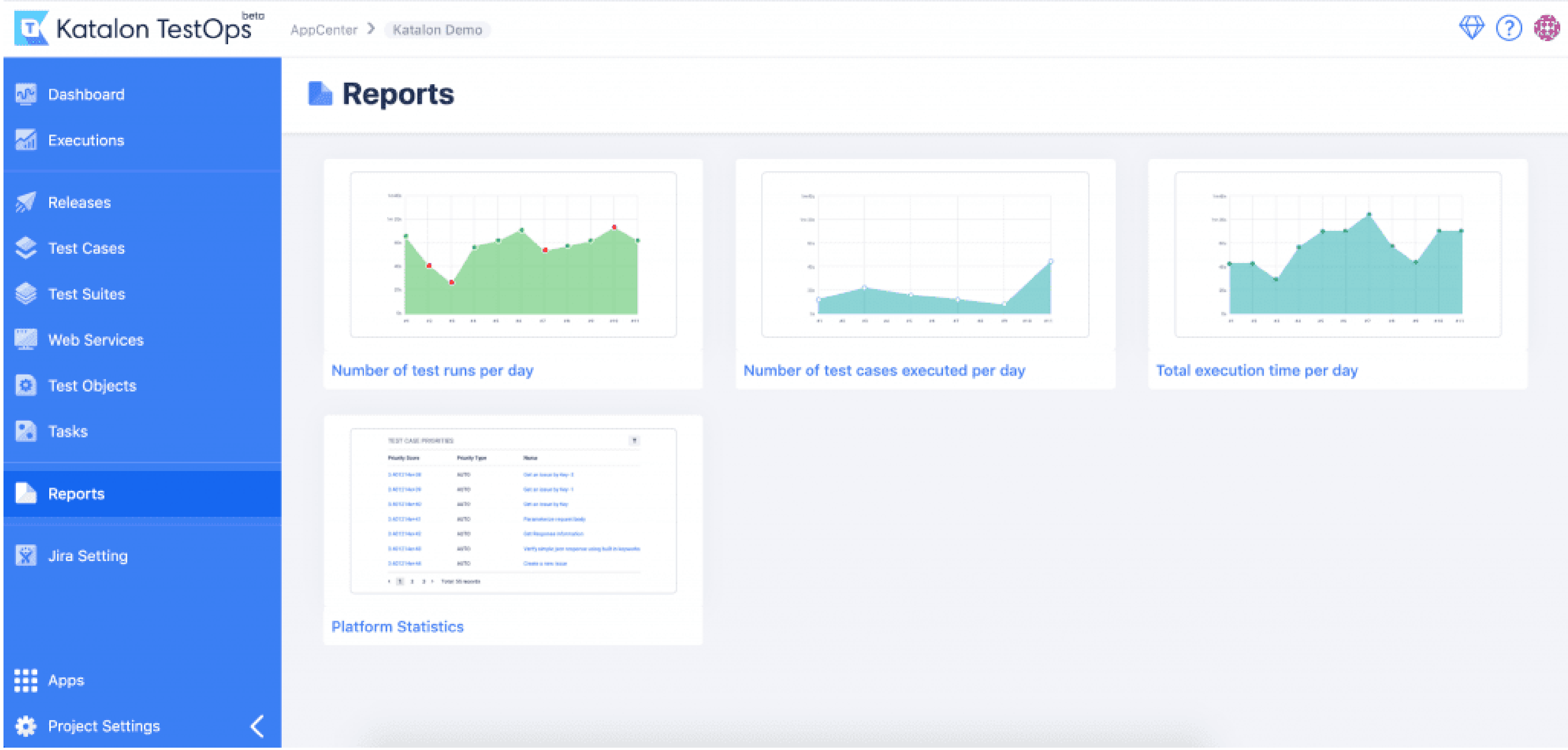
(Test case generated by Selenium IDE)

5.4 Built-in Keywords

- Katalon Studio is released with pre-defined sets of commonly used keywords or actions, users will find them enough to start implementing most test cases.
- However, if users have more advanced or specific needs, they still have a way to create custom keywords for their projects. Custom keywords are an extension of built-in keywords. They can be used the same way as built-in keywords and can be shared between users.
- Selenium and Appium: Users have to build common and reusable actions on their own in forms of programming functions.

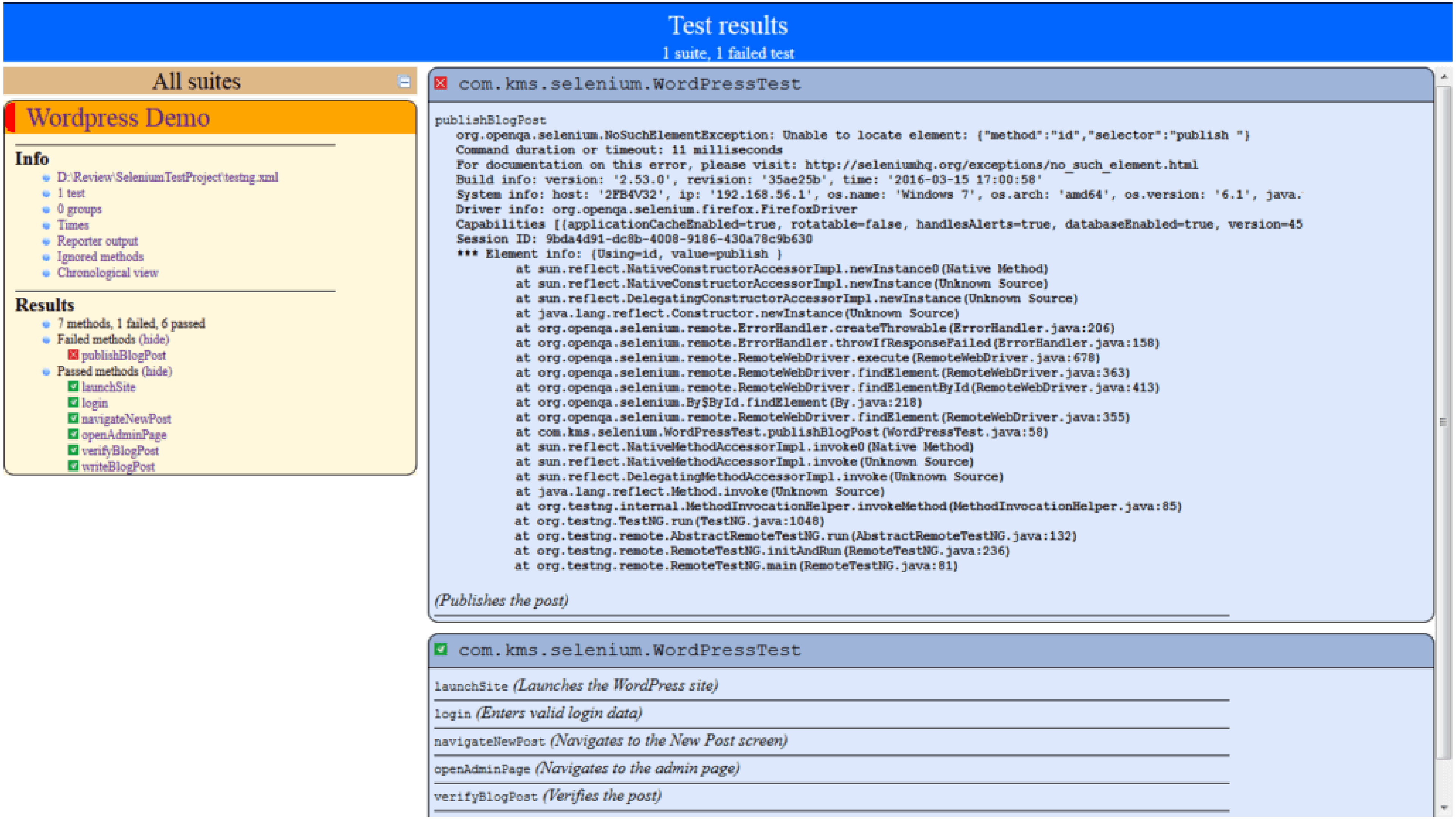
6. Reporting and Analytics

- Katalon Studio provides intuitive reports with friendly and readable messages and captured screenshots at the time of failure which can be exported into CSV, HTML and PDF files.
- Katalon TestOps (beta) visualizes test execution results in an actionable fashion that enables teams to quickly diagnose and have better insights of their automation tests.



(Katalon TestOps dashboard with visualized execution history)

- Selenium and Appium do not have reporting tools themselves, but based on the testing framework being used will have some simple report templates. Sophisticated reporting and analytics capabilities like what Katalon offers are probably among the most wanted features for Selenium and Appium.



☒ com.kms.selenium.WordPressTest

```
publishBlogPost
   org.openqa.selenium.NoSuchElementException: Unable to locate element: {"method":"id","selector":"publish "}
   Command duration or timeout: 11 milliseconds
   For documentation on this error, please visit: http://seleniumhq.org/exceptions/no_such_element.html
   Build info: version: '2.53.0', revision: '35ae25b', time: '2016-03-15 17:00:58'
   System info: host: '2FB4V32', ip: '192.168.56.1', os.name: 'Windows 7', os.arch: 'amd64', os.version: '6.1', java.
   Driver info: org.openqa.selenium.firefox.FirefoxDriver
   Capabilities [{applicationCacheEnabled=true, rotatable=false, handlesAlerts=true, databaseEnabled=true, version=45
   Session ID: 9bda4d91-dc8b-4008-9186-430a78c9b630
   *** Element info: {Using=id, value=publish }
           at sun.reflect.NativeConstructorAccessorImpl.newInstance0(Native Method)
           at sun.reflect.NativeConstructorAccessorImpl.newInstance(Unknown Source)
           at sun.reflect.DelegatingConstructorAccessorImpl.newInstance(Unknown Source)
           at java.lang.reflect.Constructor.newInstance(Unknown Source)
           at org.openqa.selenium.remote.ErrorHandler.createThrowable(ErrorHandler.java:206)
           at org.openqa.selenium.remote.ErrorHandler.throwIfResponseFailed(ErrorHandler.java:158)
           at org.openqa.selenium.remote.RemoteWebDriver.execute(RemoteWebDriver.java:678)
           at org.openqa.selenium.remote.RemoteWebDriver.findElement(RemoteWebDriver.java:363)
           at org.openqa.selenium.remote.RemoteWebDriver.findElementById(RemoteWebDriver.java:413)
           at org.openqa.selenium.By$ById.findElement(By.java:218)
           at org.openqa.selenium.remote.RemoteWebDriver.findElement(RemoteWebDriver.java:355)
           at com.kms.selenium.WordPressTest.publishBlogPost(WordPressTest.java:58)
           at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
           at sun.reflect.NativeMethodAccessorImpl.invoke(Unknown Source)
           at sun.reflect.DelegatingMethodAccessorImpl.invoke(Unknown Source)
           at java.lang.reflect.Method.invoke(Unknown Source)
           at org.testng.internal.MethodInvocationHelper.invokeMethod(MethodInvocationHelper.java:85)
           at org.testng.TestNG.run(TestNG.java:1048)
           at org.testng.remote.AbstractRemoteTestNG.run(AbstractRemoteTestNG.java:132)
           at org.testng.remote.RemoteTestNG.initAndRun(RemoteTestNG.java:236)
           at org.testng.remote.RemoteTestNG.main(RemoteTestNG.java:81)
```

(Publishes the post)

☑ com.kms.selenium.WordPressTest

launchSite *(Launches the WordPress site)*

login *(Enters valid login data)*

navigateNewPost *(Navigates to the New Post screen)*

openAdminPage *(Navigates to the admin page)*

verifyBlogPost *(Verifies the post)*

(TestNG Report with programming stack-trace, no screenshot of failure)

7. Integration

- Katalon Studio can integrate with several test management systems and bug tracking systems including Jenkins, JIRA or Test to submit bugs, defects, test logs and test results. Katalon Studio provides native plugins to connect with CI/CD applications like CircleCI, Azure DevOps, and Bamboo. For those who perform testing on the cloud or virtual machines, Katalon Studio can also integrate with SauceLabs, BrowserStack or Kobiton, a cloud-based mobile lab for cloud-based mobile testing (https://kobiton.com).
- Selenium and Appium: no integration, users have to build the integration capabilities themselves.

8. CI Integration

- Katalon Studio provides a command line interface, allowing CI Systems to call and trigger test execution.
- Selenium and Appium: users have to create and configure execution jobs (Ant tasks, maven goals, etc.) to trigger test execution.

9. Customer Support

- The Katalon team actively supports Katalon Studio's users more regular than the Selenium and Appium communities, via its website and forum.
- Selenium and Appium are open source, and users may find online help somewhere from other users or the Selenium group but nothing is guaranteed that they will get help quickly.

10. Bugs Fixes, Release Schedule

- The Selenium and Appium groups do not have a clear release plan, and bugs reported do not usually get fixed quickly.
- Katalon Studio is released monthly (minor releases for hot fixes are often available within a few days to a week), and bugs are usually fixed within days. User feedback is highly appreciated and resolved quickly.

11. Solutions for several automated testing problems

- Selenium can only deal with HTML content inside a browser. Special and customized controls such as Windows popups and embedded objects (Flash, Flex, Media) will make Selenium fail to handle.
- Katalon Studio, on the other hand, provides capabilities to handle these Windows controls and recognize images.
- Katalon Studio allows starting test from any step with an already opened session which is very helpful for troubleshooting test.
- Katalon Studio overrides Firefox Driver to make new Firefox versions (47+) backward compatible with old Web Driver while the Selenium community is still struggling with the unstable Gecko driver.

Related Articles

PRODUCTS | 6 MIN READ

AI Powered Efficiency - Katalon Offers Native Integration with Applitools

PRODUCTS | 6 MIN READ

PRODUCTS | 7 MIN READ

Eliminate Object Locator Flakiness with Katalon's New Self-healing Feature

Native Windows Recorder – Accelerate Desktop Application Testing with...

Your automation journey starts here.

Get Started

Contact Sales

Katalon

1776 Peachtree Street NW, Suite 200N, Atlanta, GA 30309
Phone: +1 678-500-9185
General & Legal inquiries: info@katalon.com
License & Purchase inquiries: business@katalon.com
Partnership inquiries: partner@katalon.com

FOLLOW US

PRODUCTS

- Katalon Studio
- Katalon Recorder
- Katalon TestOps
- Katalium
- Katalon Store
- Pricing

SOLUTIONS

- Web Testing
- API Testing
- Mobile Testing
- Desktop Testing
- Integrations
- Methodologies

SUPPORT

- Help Center
- Docs
- Community
- Repository
- Webinars

COMPANY

- About Katalon
- Blog
- Partners
- Customer Stories
- Careers

Training

Terms Privacy Policy License Agreement