# How to use Selenium WebDriver in Katalon Studio

[Katalon Studio](https://www.katalon.com/) uses the Selenium engine for automated testing of Web applications. This means that Katalon uses the Selenium WebDriver framework and hides the complexities of dealing with WebDriver from users. We don't have to care about WebDriver but instead work with Katalon keywords only.

However, We can still use WebDriver directly if we need more flexibilities than Katalon provides, we can simply write scripts working with WebDriver directly.

## How Katalon Studio uses Selenium WebDriver

We don’t need to create a webdriver instance. Katlon does it when we call **WebUI.**openBrowser keyword.

**#Selenium script**

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.firefox.FirefoxDriver;

public class Gmail\_Login {

public static void main(String[] args) {

// objects and variables instantiation

WebDriver driver = new FirefoxDriver();

**#Selenium script in Katalon:**

import org.openqa.selenium.WebDriverimport org.openqa.selenium.chrome.ChromeDriverimport com.kms.katalon.core.webui.keyword.WebUiBuiltInKeywords as WebUI

//Object instantiation in Katalon

WebUI.openBrowser('')

WebDriver driver = new ChromeDriver()

**DriverFactory Library:**

Katalon developers were aware that WebDriver methods may be useful for users, so they introduced the DriverFactory library. This library is responsible for manipulation with the WebDriver instance and offers a few useful methods for using WebDriver in Katalon.

**ChangeWebDriver() : Method of DriveFactory Liberary:**

It is not possible to use Katalon keywords with custom Webdriver instance by default,so, for changing the webdriver instance to default one,we need changeWebDriver() method that allow us to use full functionality of katalon.

import org.openqa.selenium.WebDriverimport org.openqa.selenium.chrome.ChromeDriverimport com.kms.katalon.core.webui.keyword.WebUiBuiltInKeywords as WebUI

//Set property for webdriver

System.setProperty("webdriver.chrome.driver", "C:\\test\\chromedriver.exe") WebDriver driver = new ChromeDriver()

// this won't work and exception would be thrown

 WebUI.click(testObject)

 DriverFactory.changeWebDriver(driver)// change the webdriver to default driver

 // now, it is working as expected because instance has been changed to default instance of katalon

 WebUI.click(testObject)

getWebDriver() : **Method of DriveFactory Liberary:**

Through this method we can get instance of webdriver created by Katalon and use Webdriver method after that.

WebUI.openBrowser('')

//Using this method,We can get a WebDriver instance created in the openBrowser() method.

WebDriver myDriver = DriverFactory.getWebDriver()

Transition from Java Selenium projects to Katalon’s

If We already have Selenium projects written in Java, it is easy to start using Katalon without losing your previous work.All Java code would work also in Katalon, We should only call the changeWebDriver() method to be able to use Katalon's built-in keywords.

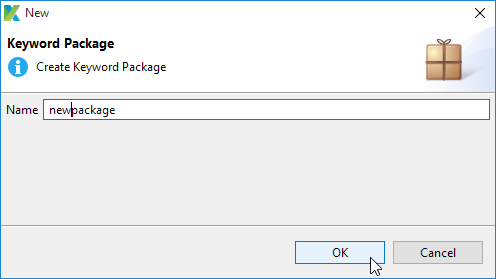
**Use Selenum Webdriver script as custom keywords in Katalon**

**Custom Keywords:**

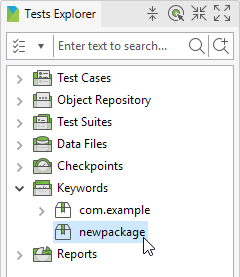
We can define custom keywords to extend the capabilities of Katalon Studio. Once created, custom keywords can be used when implementing test cases, just like other built-in keywords.

**Create new Custom keyword:**

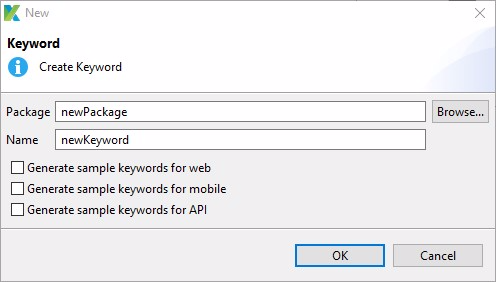
1. Select **File > New > Package** from the main menu. The **New Keyword Package** dialog is displayed. Enter a name for your package and click **OK**.



2. A new package is created under the **Keywords** folder in Test Explorer accordingly.



3. By default, a class name cannot start with a number, contain spaces, or have special characters. The Java naming convention suggests creating a class name using a noun or a noun phrase, with the first letter of each word capitalized, to better manage the project.



You can generate sample custom keywords for Web, Mobile, and API Testing. Refer to [this guide](https://docs.katalon.com/katalon-studio/docs/sample-custom-keywords.html).

For complete reference or creating custom keyword follow given link: <https://docs.katalon.com/katalon-studio/docs/introduction-to-custom-keywords.html#create-a-custom-keyword>

A new keyword is created under the specified **package** accordingly.

Import packages for webdrivers or Katalon will automatically ask for importing this pacakages while creating webdriver code.

import org.openqa.selenium.WebDriver

import org.openqa.selenium.WebElement

import org.openqa.selenium.By.ByXPath

import com.kms.katalon.core.webui.driver.DriverFactory

import org.openqa.selenium.By

Following keyword created through @Keyword annotation.

Example: Login in facebook account using custom keyword.

def logInFacebook( ){

//open browser

WebUI.openBrowser("www.facebook.com");

//instantiate webdriver object

WebDriver driver = DriverFactory.getWebDriver()

//Locate element and put the value

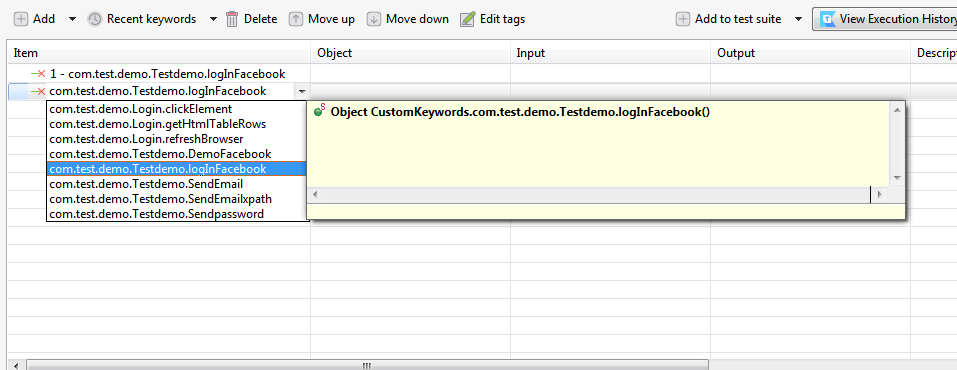
[driver.findElement(By.xpath("//input[@id='email']")).sendKeys("xyz](mailto:driver.findElement(By.xpath(\"/input[@id='email']\")).sendKeys(\"xyz) @gmail.com");

//locate element and put the value

driver.findElement(By.xpath("//input[@id='pass']")).sendKeys("123456");

driver.findElement(By.xpath("//button[@id='u\_0\_b']")).click();

}

Call is custom keywords as: Go to test case >> Click on Add>>Custom keyword

Groovy </>script will look like :

import static com.kms.katalon.core.checkpoint.CheckpointFactory.findCheckpoint

import static com.kms.katalon.core.testcase.TestCaseFactory.findTestCase

import static com.kms.katalon.core.testdata.TestDataFactory.findTestData

import static com.kms.katalon.core.testobject.ObjectRepository.findTestObject

import static com.kms.katalon.core.testobject.ObjectRepository.findWindowsObject

import com.kms.katalon.core.checkpoint.Checkpoint as Checkpoint

import com.kms.katalon.core.cucumber.keyword.CucumberBuiltinKeywords as CucumberKW

import com.kms.katalon.core.mobile.keyword.MobileBuiltInKeywords as Mobile

import com.kms.katalon.core.model.FailureHandling as FailureHandling

import com.kms.katalon.core.testcase.TestCase as TestCase

import com.kms.katalon.core.testdata.TestData as TestData

import com.kms.katalon.core.testobject.TestObject as TestObject

import com.kms.katalon.core.webservice.keyword.WSBuiltInKeywords as WS

import com.kms.katalon.core.webui.keyword.WebUiBuiltInKeywords as WebUI

import com.kms.katalon.core.windows.keyword.WindowsBuiltinKeywords as Windows

import internal.GlobalVariable as GlobalVariable

import org.openqa.selenium.WebDriver as WebDriver

import org.openqa.selenium.chrome.ChromeDriver as ChromeDriver

CustomKeywords.'com.test.demo.Testdemo.logInFacebook'()

**Conclusion:**

The main advantage of this approach is that you can benefit from both sides – Katalon and WebDriver. If something is difficult to do in Katalon – do it in WebDriver!

**References**:

How to use selenium webdriver in Katalon

<https://docs.katalon.com/katalon-studio/docs/using_selenium_webdriver_katalon_studio.html#how-katalon-studio-uses-selenium-webdriver>

Links for Katalon java doc link

<https://docs.katalon.com/javadoc/index.html>

Link to check methods and properties of the DriverFactory class, <https://api-docs.katalon.com/com/kms/katalon/core/webui/driver/DriverFactory.html>

Introduction to custom keyword

<https://docs.katalon.com/katalon-studio/docs/introduction-to-custom-keywords.html#create-a-package>