Topics:

* Introduction & getting started.
* Download & setup.
* Features and functionalities of the tool.
* Web UI automation testing.
* Test creation modes.
* Test case | Test suite | Test suite collections.
* Logs & Reports.
* Custom reports.
* Execution environments.
* Katalon recorder.
* Record & Spy on existing browser.
* Variables.
* Data driven testing.
* Checkpoints.
* Command line.
* Failure handling
* Listeners.
* Setup & Tear down
* Self-healing
* Debugging / Troubleshooting
* Frames handling
* Visual validation
* How to integrate with GIT (SCM)

🡪The official website of katalog studio is <https://katalon.com/>.

1. What is Katalon studio.
2. Why do we use it.
3. What can you use katalon studio for.

Katalon studio is a platform that you can use for automation testing on web, mobile API and desktop and you can use it for any of these automation testing.

It is a very simple and easy tool to use.

Download and Setup:

1. Tool prerequisite.
2. Hardware / software configurations.
3. How to setup katalon studio.
4. Setup on windows.

System Requirements - https://docs.katalon.com/katalon-studio/docs/supported-environments.html

Step 1: Go to https://www.katalon.com/.

Step 2: Create an account - Sign Up

Step 3: Sign In

Step 4: Click on Download

Step 5: Download Katalon Studio & Katalon Runtime Engine

Step 6: Unzip the folders and keep at any location on your system

Step 7: Open Katalon Studio folder and click on the katalon.exe

Step 8: Check Katalon Studio is started.

Explore GUI and features:

Create Project

Step 1: Go to menu File > New > Project.

Step 2: Provide Project name, type, location > create the project

Tour of Katalon Studio GUI

Features and Options

Settings and Preferences

Create Automation tests:

How to create automation tests in Katalon Studio

Different ways to create Automation tests.

Step 1: Create a new test case.

File > New > Test Case

OR

Test Explorer > Test Cases > Rt Click > New > Test Case

Step 2: Provide test case name (Description & Tags are optional)

Step 3: Click OK

Modes in Katalon Studio:

How to create tests in Record mode

How to create tests in Manual mode

How to create tests in Script mode

1. Recording Mode

Step 1: On the main toolbar click on Web Recorder

Step 2: Add the URL and select the browser.

Step 3: Record the steps.

Step 4: Stop Recording

Step 5: Save the steps and objects.

Step 6: Save the test case and execute.

Step 7: Verify the execution.

1. Manual Mode

Step 1: Click on Spy Web

Step 2: Provide application URL and select the browser.

Step 3: Capture the objects.

Right click on Object and select Capture Object OR

Hover over the object and once it is highlighted, Press the key combination OR

Click on the object and once it is highlighted, Press the key combination OR

Click and keep the mouse pressed and press the key combination.

Step 4: Click on Save button and save the objects.

Step 5: Check all the objects are present under Object Repository

Raghav Pal

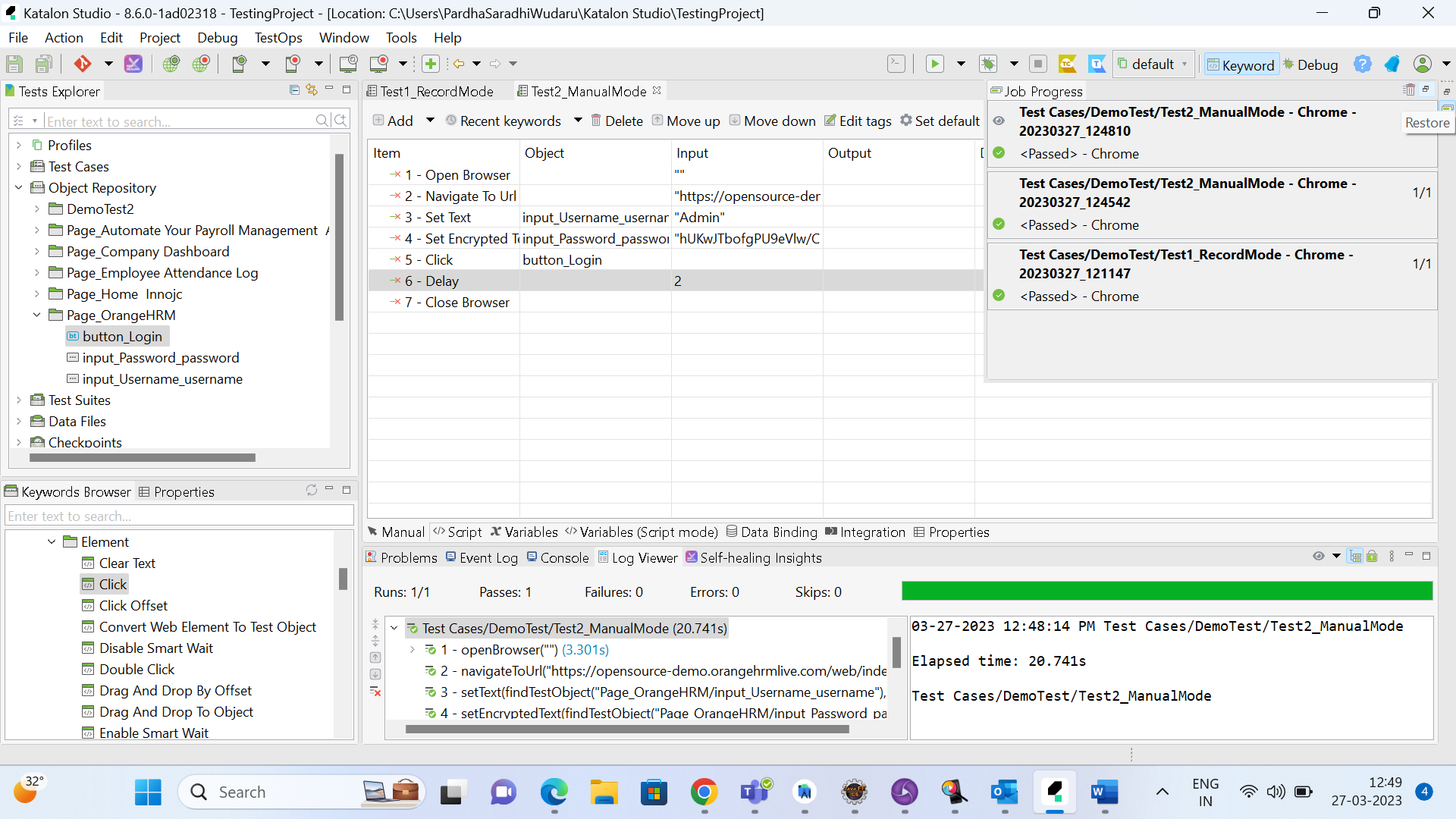
Automation Step by Step

Step 6: Create a new test case.

Step 7: Add steps and objects to the test case.

Step 8: Save

Step 9: Run and verify.



1. Script Mode:

Step 1: Click on Spy Web

Step 2: Provide application URL and select the browser.

Step 3: Capture the objects.

Right click on Object and select Capture Object OR

Hover over the object and once it is highlighted, Press the key combination OR

Click on the object and once it is highlighted, Press the key combination OR

Click and keep the mouse pressed and press the key combination

Step 4: Click on Save button and save the objects.

Step 5: Check all the objects are present under Object Repository

Step 6: Create a new test case.

Step 7: Go to the script section of the test case

Step 8: Create scripts for test steps.

e.g.

WebUI.openBrowser('')

WebUI is the class that contains all the action methods.

Step 8: Save project.

Step 9: Run and verify.

Graphical user interface, text, application

Description automatically generated

Test Suite:

What is Test Suite

How to create Test Suite

Test Suite features & execution

Test Suite - A collection of test cases

Step 1: Create a new Test Suite

Step 2: Add test cases to the suite.

Step 3: Check the test cases for execution.

Step 4: Update Execution Information as needed

Step 5: Save and Run

Test Suite Collections:

What is Test Suite Collection

How to create Test Suite collection

Test Suite collection features & execution

Test Suite Collection - A collection of Test Suites

Step 1: Create a new Test Suite Collection

File > New > Test Suite Collection

OR

Test Explorer > Test Suites > Rt Click > New > Test Suite Collection

Step 2: Add test suites to the collection.

Step 3: Setup Execution Information as needed.

Step 4: Save and Run

Run Configuration: To add extra information to execute with the selected environment.

For example: Select mobile devices to be executed for Android environment.

Logs & Reports

Different types of reports

Test Case Logs

Test Suite report

Test Suite collection report

Logs get generated for following executions.

Test Case

Test Suite

Test Suite Collections

Reports get generated for following executions.

Test Suite

Test Suite Collections

Custom keywords:

What are custom keywords.

How to create custom keywords.

Built-In Keywords

Custom Keywords

User defined keywords, that can be scripted and exposed as keywords to be used in Test Cases

Step 1: Create a new Custom Keyword

File > New > Keyword

OR

Test Explorer > Keywords > Rt click > New > Keyword.

Step 2: Provide a keyword name.

Step 3: In the editor script for your keyword functions

Step 4: Provide annotation the function - @Keyword.

@Keyword

def hello () {

println " Hello World... "

}

Step 5: Save and Use the keyword in a Test Case

Drag and drop the keyword from Keywords Browser

OR

Click on Add > Custom Keyword > Select the keyword.

Step 6: Save & Run

@Keyword

def helloUser(String user){

println " Hello "+user

}

Can declare the category of the keyword using keyword Object.

@Keyword (keywordObject = "Browser")

def refreshBrowser() {

}

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.annotation.Keyword

import com.kms.katalon.core.checkpoint.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

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

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

import com.kms.katalon.core.testobject.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

public class MyKeywords {

@Keyword(KeywordObject = "hello")

def hello() {

println("Hello world")

}

@Keyword

def login() {

WebUI.setText(findTestObject('Object Repository/Page\_OrangeHRM/input\_Username\_username'), 'Admin')

WebUI.setText(findTestObject('Object Repository/Page\_OrangeHRM/input\_Password\_password'), 'admin123')

WebUI.click(findTestObject('Object Repository/Page\_OrangeHRM/button\_Login'))

}

@Keyword(KeywordObject = "hello")

def helloUser(String user) {

println " Hello "+user

}

}