TESTBITS

KATALON STUDIO TRAINING

Presented By : Testbits Sdn Bhd

Mini Assessment 1 Duration (30mins)

- Create and Automate Test Cases on CMS Katalon Shop based on the test cases given.

LEARNING OUTCOME

- Participants able to implement test cases creation based on situation given.
- Participants able to create TS / TSC and understand their features.
- Participants able to generate various type of reports.
- Participants able to setup email notifications for TS / TSC reporting.
- Participants able to create their own custom keywords to be used in their respective TCs.

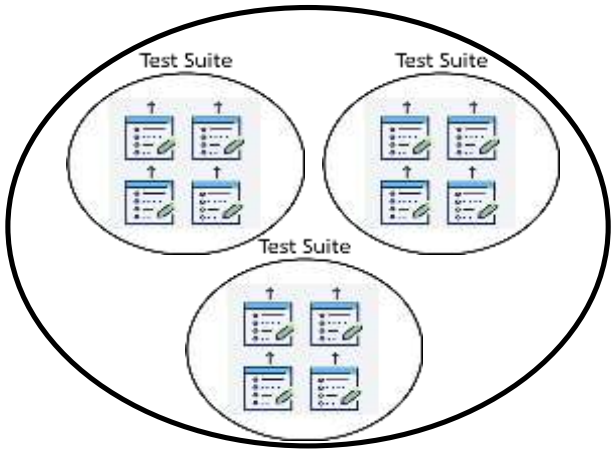
2.1 TEST SUITE & TEST SUITE COLLECTION

Test Suite - A collection of test cases or test scenarios that are designed to verify the functionality and quality of a software application. It includes a set of inputs, expected outputs, and the conditions under which the software should behave.

Test Suite Collection - A test suite collection is a group of related test suites that are designed to test an entire software application or a specific module or component of the application.

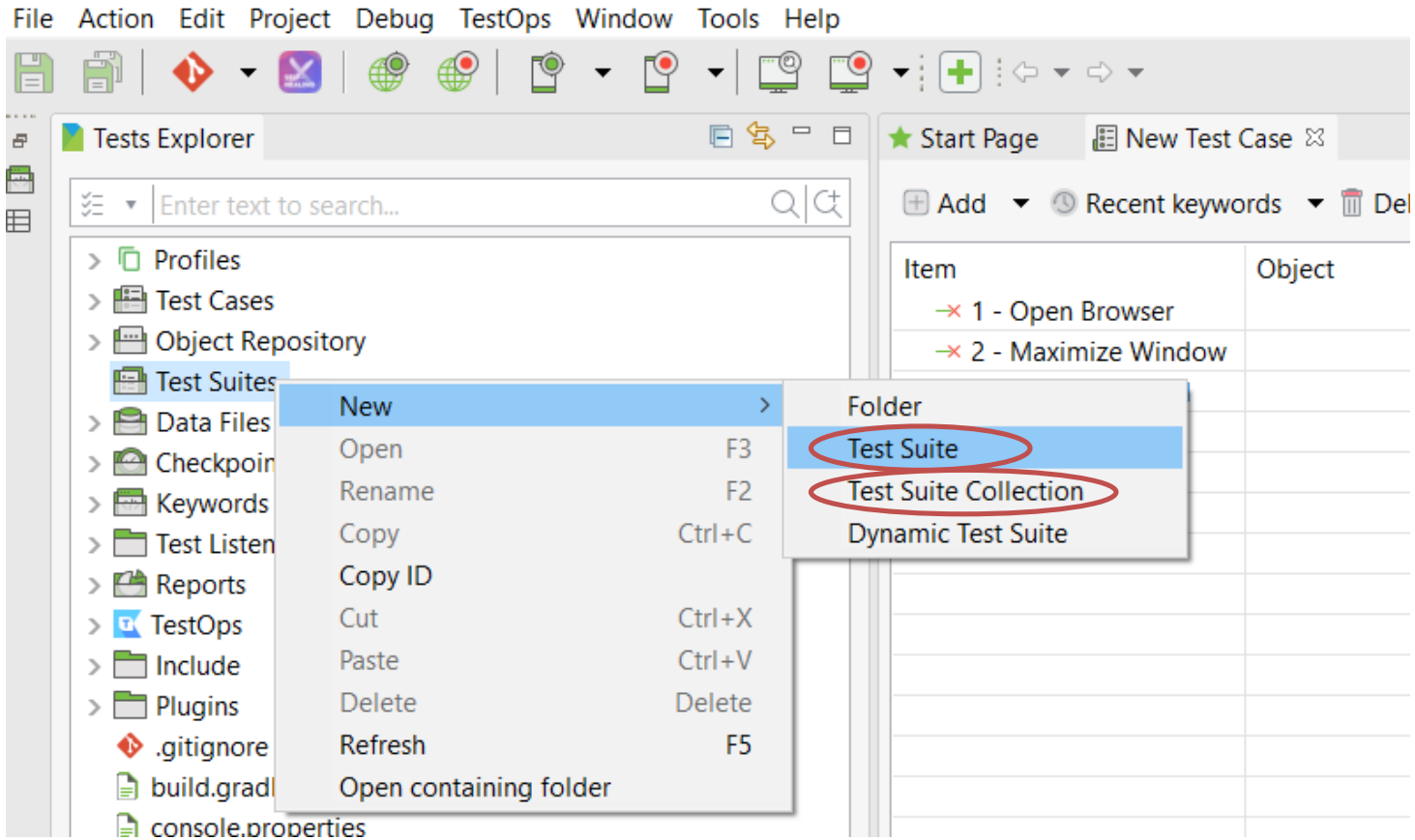
Test Suite

Test Suite Collection



2.1 TEST SUITE & TEST SUITE COLLECTION

Step 1: Create New Test Suite/ Test Suite Collection



1. Right Click Test Suite Folder
2. Select New
3. Click Test Suite or Test Suite Collection

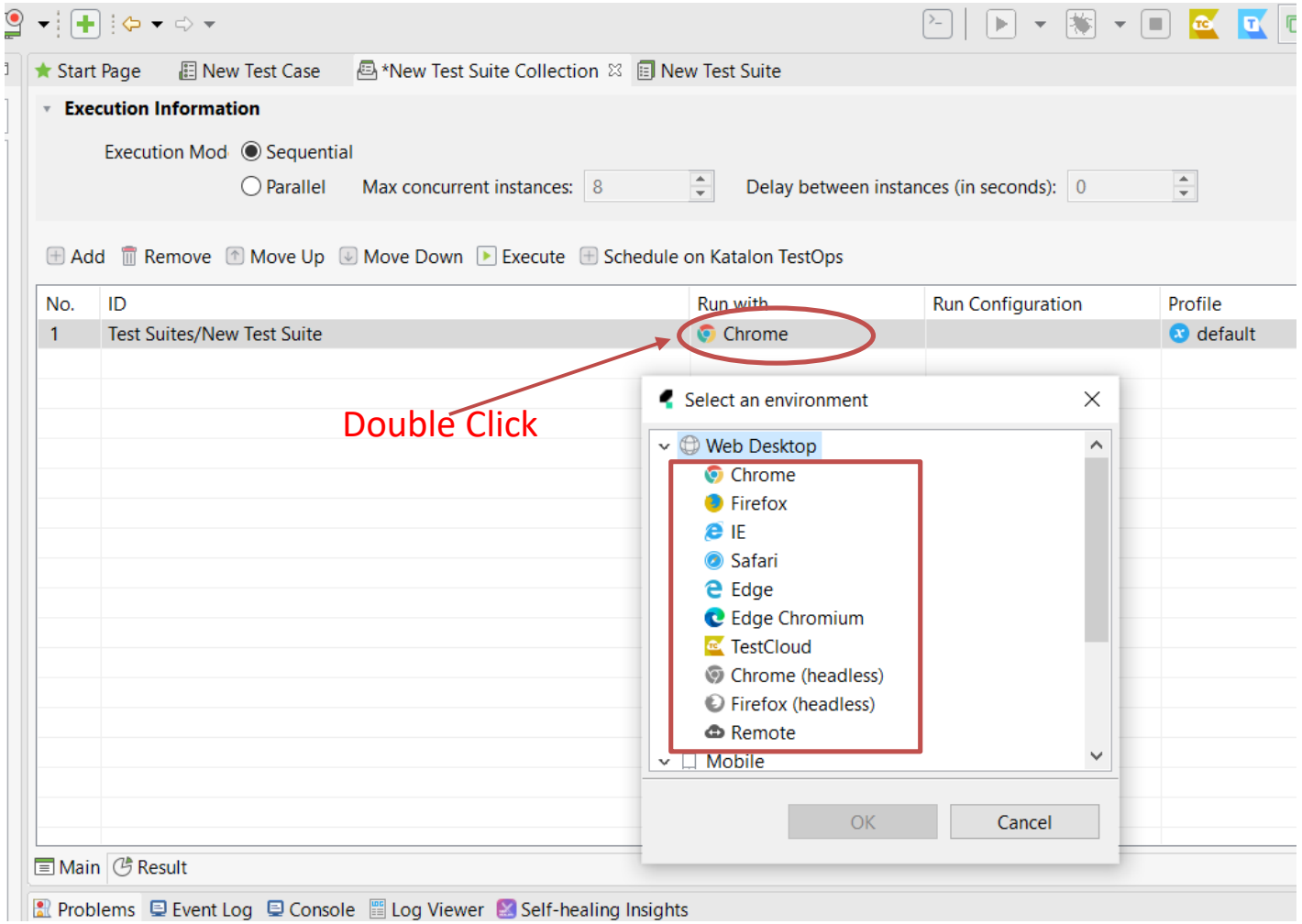
2.1 TEST SUITE & TEST SUITE COLLECTION

Step 2: Add Test Case into Test Suite

Step 3: Add Test Suite into Test Suite Collection

Step 4: Execute Test Suite and Test Suite Collection

2.1.1 Multiple Browser Support (For TSC)



<u>Multi-Browser Support only available in Test Suite Collection</u>

Step 1: Create New Test Suite Collection

Step 2: Add Test Suite to Test Suite Collection

Step3: Select Browser

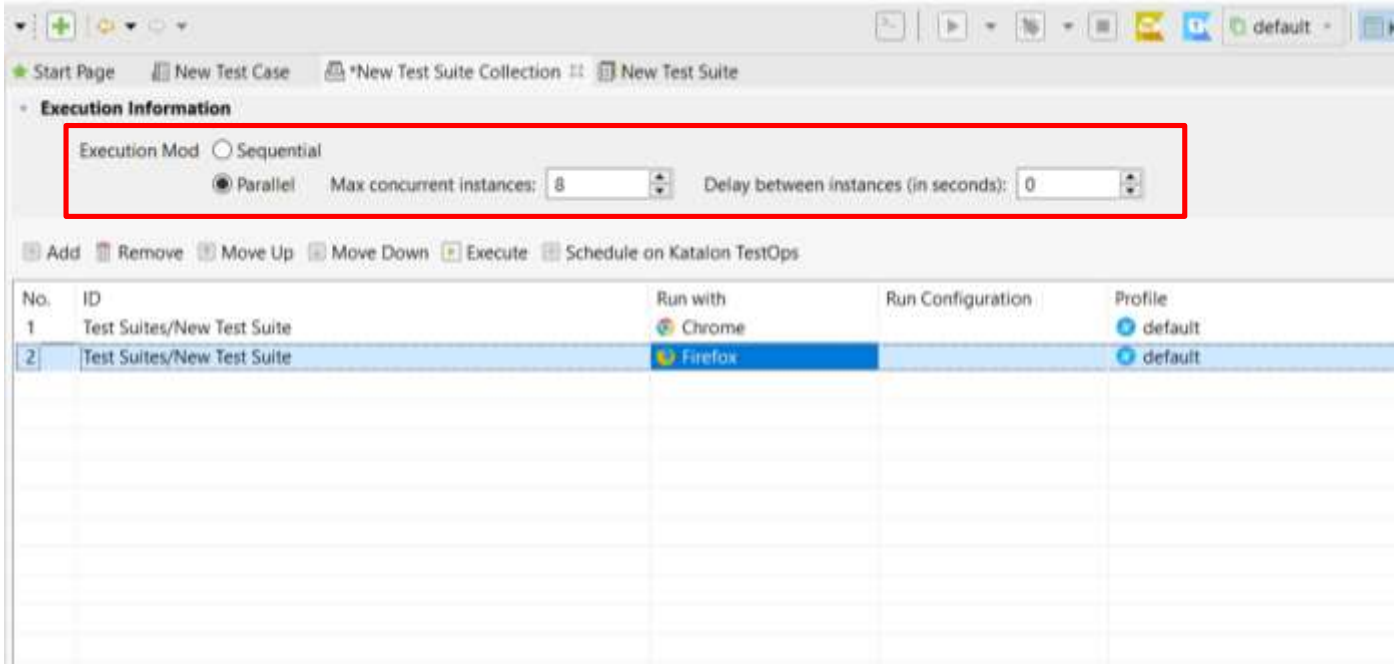
Step4:Execute Test Suite Collection

2.1.2 Sequential and Parallel (For TSC)

Sequential Execution Mod – The Test Suites will execute one-by-one according to its order.

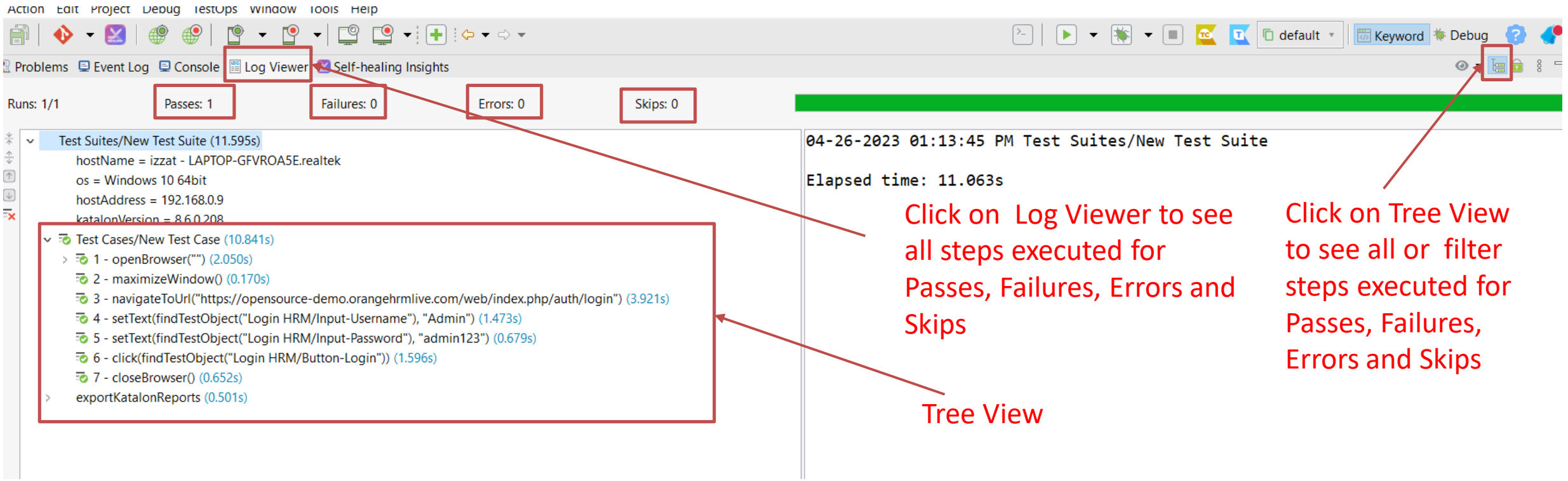
Parallel Execution Mod – All the Test Suites will execute concurrent or simultaneously.

Ex. Both Test Suite will run for both Chrome and Firefox Browser at the same time in Parallel Mod



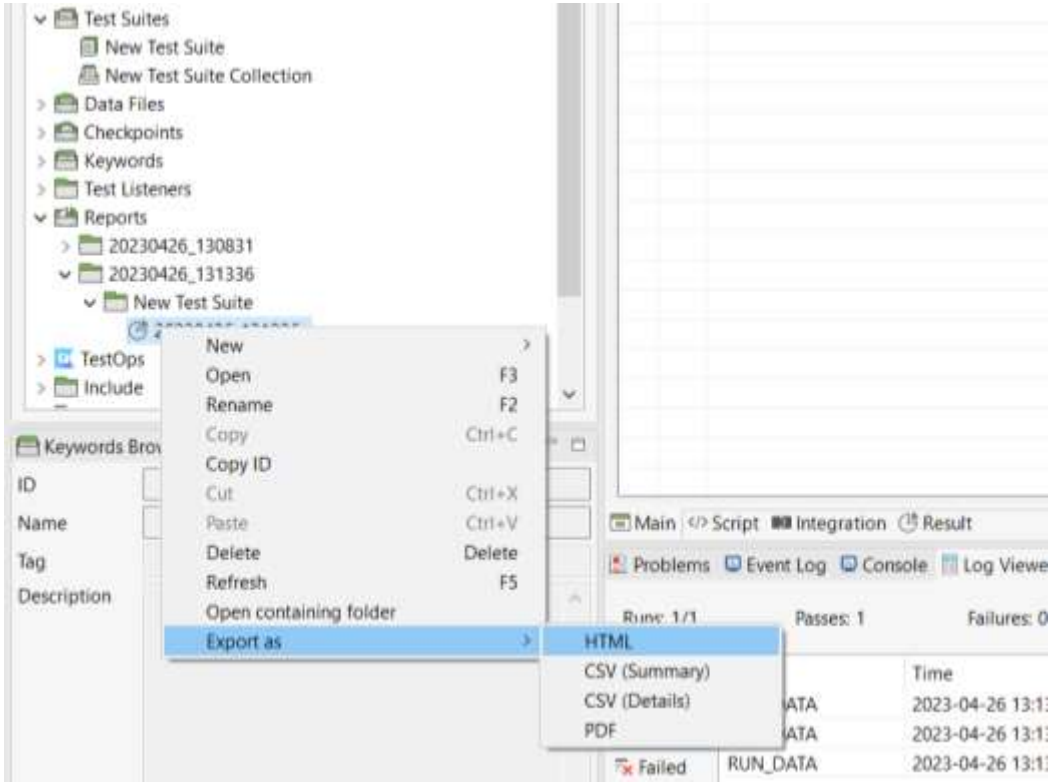
2.2 Logs & Reports

Upon executing Test Cases, Test Suite and Test Suite Collection regardless the results, Katalon Studio will generate Logs that can be viewed for test cases that Passed or Failed(Log Viewer).



2.2 Logs & Reports

Overall Reports only generated everytime Test Suite or Test Suite Collection is executed and can be exported as html, PDF and CSV Format.



Export/ View Report

Step 1: Open Reports Folder in the Test Explorer

Step 2: Right-Click on the Test Suite/Test Suite Collection Folder

Step3: Select Export as HTML,CSV or PDF

2.2 Logs & Reports

Host name: izzat - LAPTOP-GFVROA5E realtek
Local OS: Windows 10 64bit
Katalon version: 8.6.0.208
Browser: Chrome 112.0.0.0

Test Execution Log

TEST SUITE **New Test Suite**

Full Name: New Test Suite
Start / End / Elapsed: 2023-04-26 13:13:45.413 / 2023-04-26 13:13:56.476 / 00:00:11.063
Status: 1 test total, 1 passed, **0 failed**, **0 error**, 0 incomplete, 0 skipped

TEST CASE **Test Cases/New Test Case**

Full Name: New Test Suite/Test Cases/New Test Case
Start / End / Elapsed: 2023-04-26 13:13:45.635 / 2023-04-26 13:13:56.476 / 00:00:10.841
Status: PASSED

TEST STEP *openBrowser("")*

Start / End / Elapsed: 2023-04-26 13:13:45.932 / 2023-04-26 13:13:47.982 / 00:00:02.050

```
13:13:46.164    INFO   Starting 'Chrome' driver
13:13:46.203    INFO   Action delay is set to 0 milliseconds
13:13:47.982    PASSED Browser is opened with url: ''
```

TEST STEP *maximizeWindow()*

Start / End / Elapsed: 2023-04-26 13:13:47.963 / 2023-04-26 13:13:48.153 / 00:00:00.170

```
13:13:48.153    PASSED Current window maximized
```

TEST STEP *navigateToUrl("https://opensource-demo.orangehrmlive.com/web/index.php/auth/login")*

Start / End / Elapsed: 2023-04-26 13:13:48.154 / 2023-04-26 13:13:52.075 / 00:00:03.921

```
13:13:52.074    PASSED Navigate to 'https://opensource-demo.orangehrmlive.com/web/index.php/auth/login' successfully
```

TEST STEP *setText(findTestObject("Login HRM/Input-Username"), "Admin")*

Start / End / Elapsed: 2023-04-26 13:13:52.075 / 2023-04-26 13:13:53.548 / 00:00:01.473

```
13:13:53.548    PASSED Text 'Admin' is set on object 'Object Repository/Login HRM/Input-Username'
```

TEST STEP *setText(findTestObject("Login HRM/Input-Password"), "admin123")*

Start / End / Elapsed: 2023-04-26 13:13:53.548 / 2023-04-26 13:13:54.227 / 00:00:00.679

```
13:13:54.226    PASSED Text 'admin123' is set on object 'Object Repository/Login HRM/Input-Password'
```

TEST STEP *click(findTestObject("Login HRM/Button-Login"))*

Start / End / Elapsed: 2023-04-26 13:13:54.227 / 2023-04-26 13:13:55.823 / 00:00:01.596

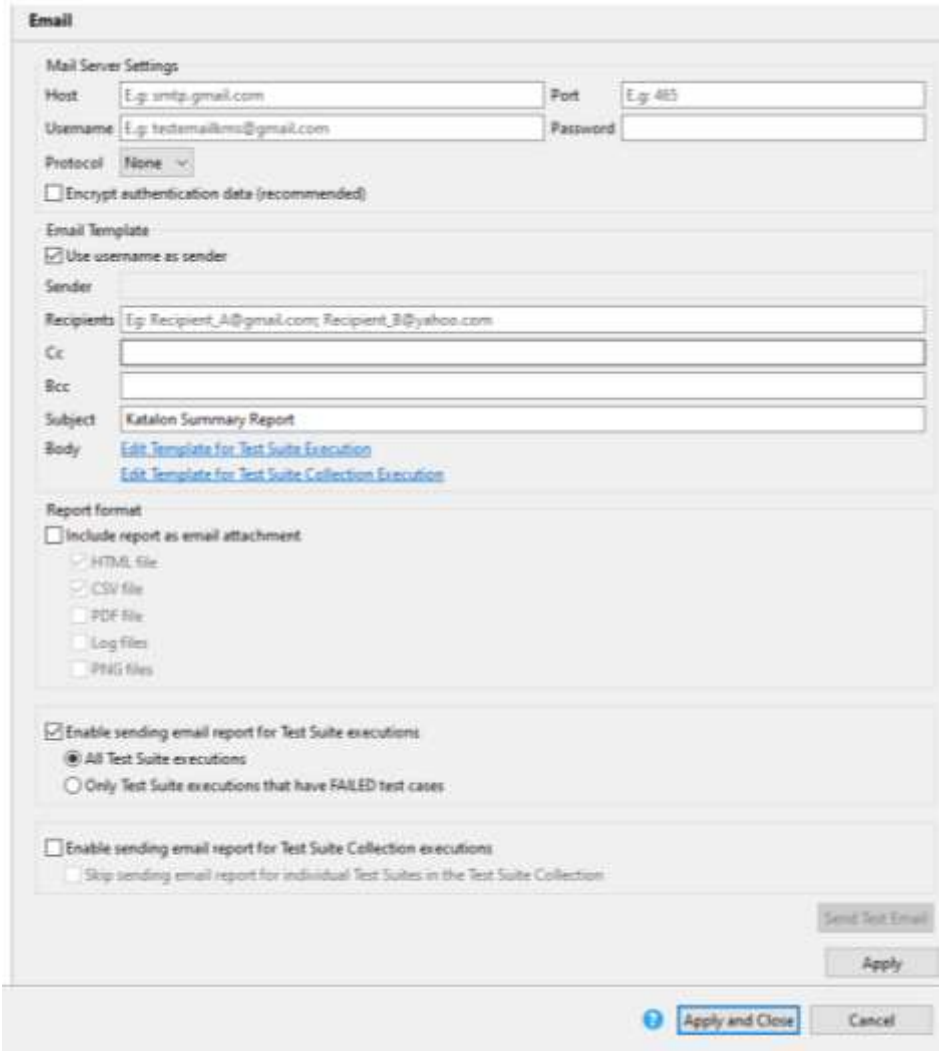
```
13:13:55.822    PASSED Object: 'Object Repository/Login HRM/Button-Login' is clicked on
```

TEST STEP *closeBrowser()*

Start / End / Elapsed: 2023-04-26 13:13:55.824 / 2023-04-26 13:13:56.476 / 00:00:00.652

```
13:13:56.475    PASSED Browser is closed
```

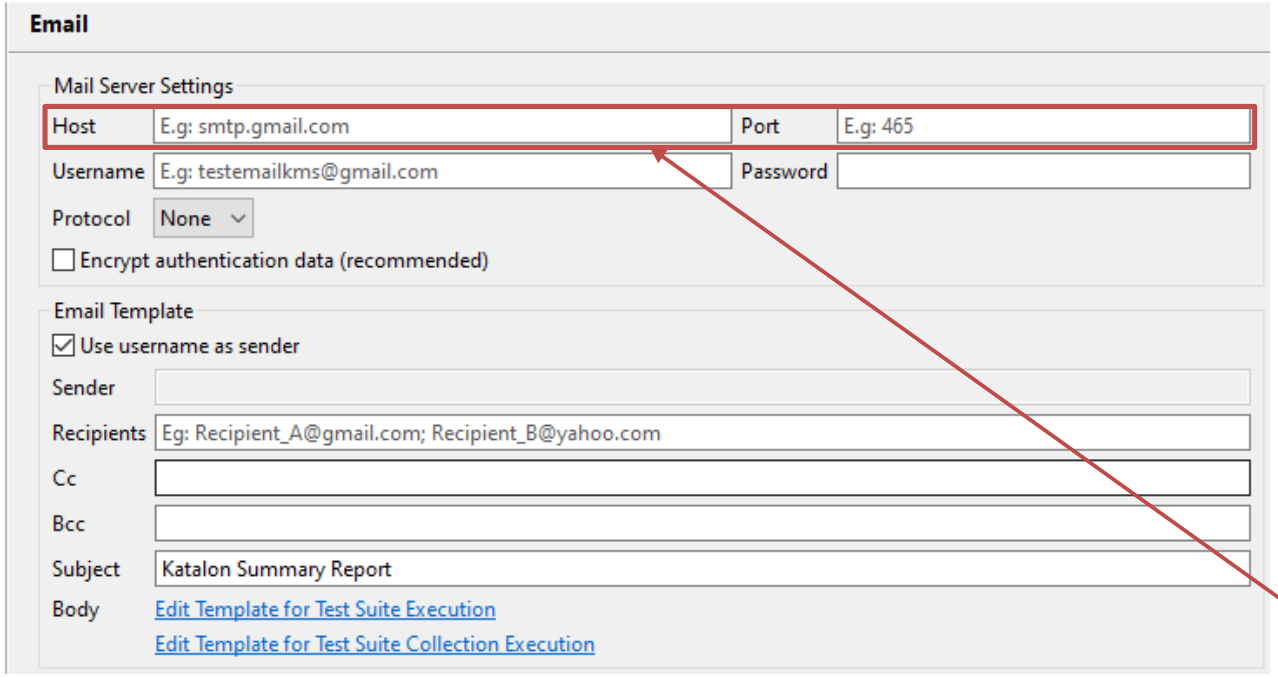
2.3 Email Notifications



After a test suite or test suite collection execution, you might want to automatically send summary reports to your own email or other stakeholders to notify them about the test result. This document shows you how to set up your mail server and customize email reports to automatically send out a summary report email whenever a test execution finishes.

configure global email settings in
Project > Settings > Email

2.3 Email Notifications



- **Username**: Your full email account to authenticate with the server (e.g., yourusername@gmail.com)

- **Password**: Your email password to authenticate with the server. This could be a password generated from App Passwords.

Email sever	Host	Port	Reference
Gmail	`smtp.gmail.com`	465 or 587	Check Gmail through other email platforms
Outlook	`smtp.office365.com`	587 or 25	How to set up a multifunction device or application to send email using Microsoft 365 or Office 365
Yahoo! Mail	`smtp.mail.yahoo.com`	465	POP access settings and instructions for Yahoo Mail

2.4 Custom Keywords

A custom keyword is a User define keyword that extends the capabilities of Katalon Studio in addition to built-in Keywords.

How to Create Custom Keywords:

Step 1: Create New Package

-Right Click at the “Keywords” Folder and select “New” and Choose “Package”

-Give a name to your package. (Make sure the Package name starts in lowercase Ex. “myCustomKeywords”)

Step 2: Create New Keyword

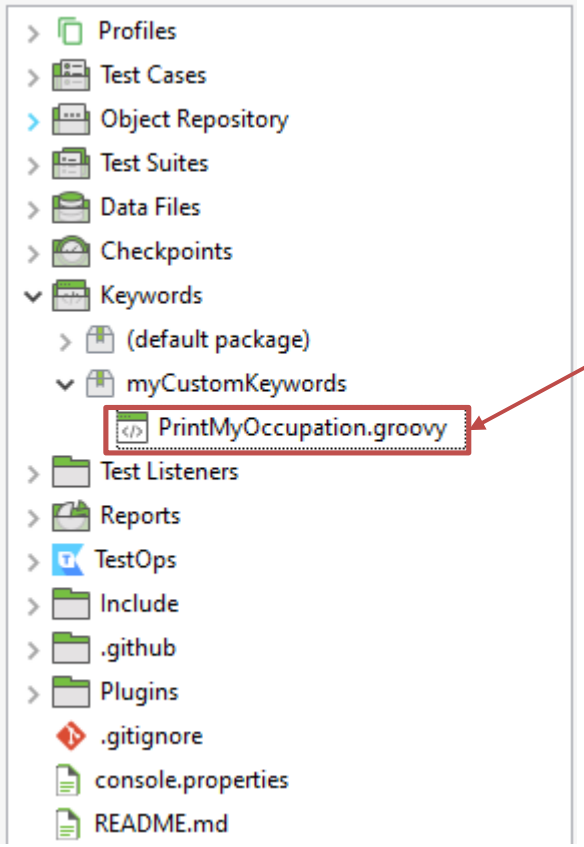
-Right Click on the newly create Package folder and select “New” and Choose “Keyword”

-Give a name to your Keyword and it will create a new .groovy script.

(Ex. PrintMyOccupation)

2.4 Custom Keywords

Step 2: Create New Key



Welcome | PrintMyOccupation.groovy

```
1  package myCustomKeywords
2
3  import static com.kms.katalon.core.checkpoint.CheckpointFactory.findCheckpoint
4  import static com.kms.katalon.core.testcase.TestCaseFactory.findTestCase
5  import static com.kms.katalon.core.testdata.TestDataFactory.findTestData
6  import static com.kms.katalon.core.testobject.ObjectRepository.findTestObject
7  import static com.kms.katalon.core.testobject.ObjectRepository.findWindowsObject
8
9  import com.kms.katalon.core.annotation.Keyword
10 import com.kms.katalon.core.checkpoint.Checkpoint
11 import com.kms.katalon.core.cucumber.keyword.CucumberBuiltinKeywords as CucumberKW
12 import com.kms.katalon.core.mobile.keyword.MobileBuiltInKeywords as Mobile
13 import com.kms.katalon.core.model.FailureHandling
14 import com.kms.katalon.core.testcase.TestCase
15 import com.kms.katalon.core.testdata.TestData
16 import com.kms.katalon.core.testobject.TestObject
17 import com.kms.katalon.core.webservice.keyword.WSBuiltInKeywords as WS
18 import com.kms.katalon.core.webui.keyword.WebUiBuiltInKeywords as WebUI
19 import com.kms.katalon.core.windows.keyword.WindowsBuiltinKeywords as Windows
20
21 import internal.GlobalVariable
22
23 public class PrintMyOccupation {
24 }
25
```

It should look like this with the Package folder and custom keyword created

2.4 Custom Keywords

Step 3: Create New Function

```groovy
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 PrintMyOccupation {

    @Keyword (keywordObject = "<category_name>")
    def keywordName(parameters…) {
    // enter your code here
    // you can use either Groovy or Java
    }



}
```

Item	Description	Required
`@Keyword`	The annotation to indicate that the block of code below is the definition of a keyword.	Yes
`keywordObject`	The category of your custom keyword (available from version 7.5.5).	No
`keywordName`	The name for your custom keyword.	Yes
parameters	The list of parameters to be used in the custom keyword.	No

2.4 Custom Keywords

Step 3: Create New F...

```groovy
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 PrintMyOccupation {

    @Keyword (keywordObject = "PrintFunction")
    def SelectOccupation (int input) {

        switch (input) {

            case 1:
            println("Automation Tester");
            break;

            case 2:
            println("Manual Tester");
            break;

            case 3:
            println("Automation and Manual Tester");
            break;


        }
    }
```

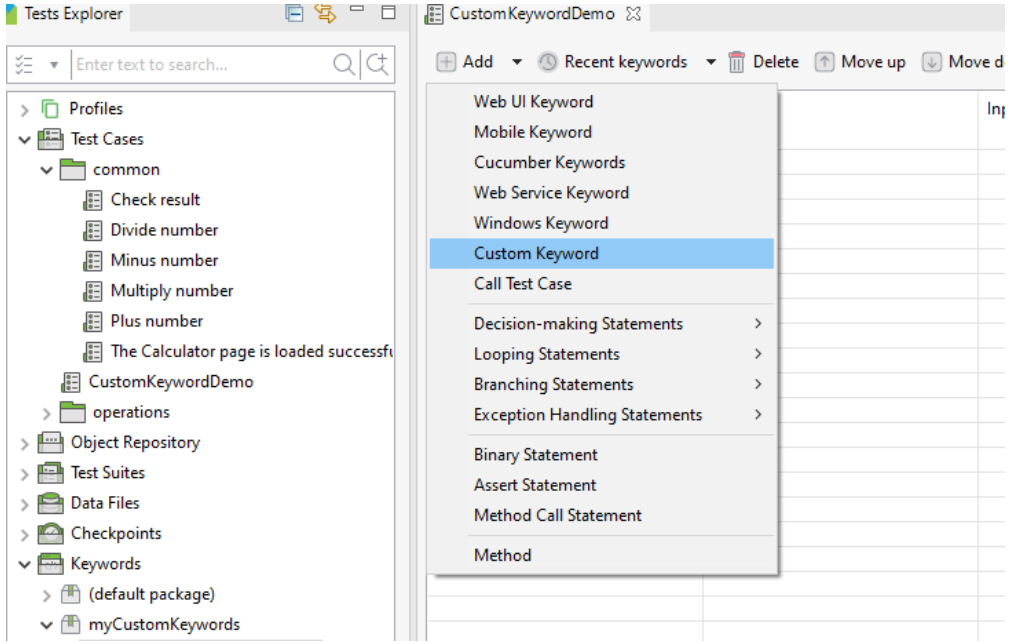
Example to print Occupation

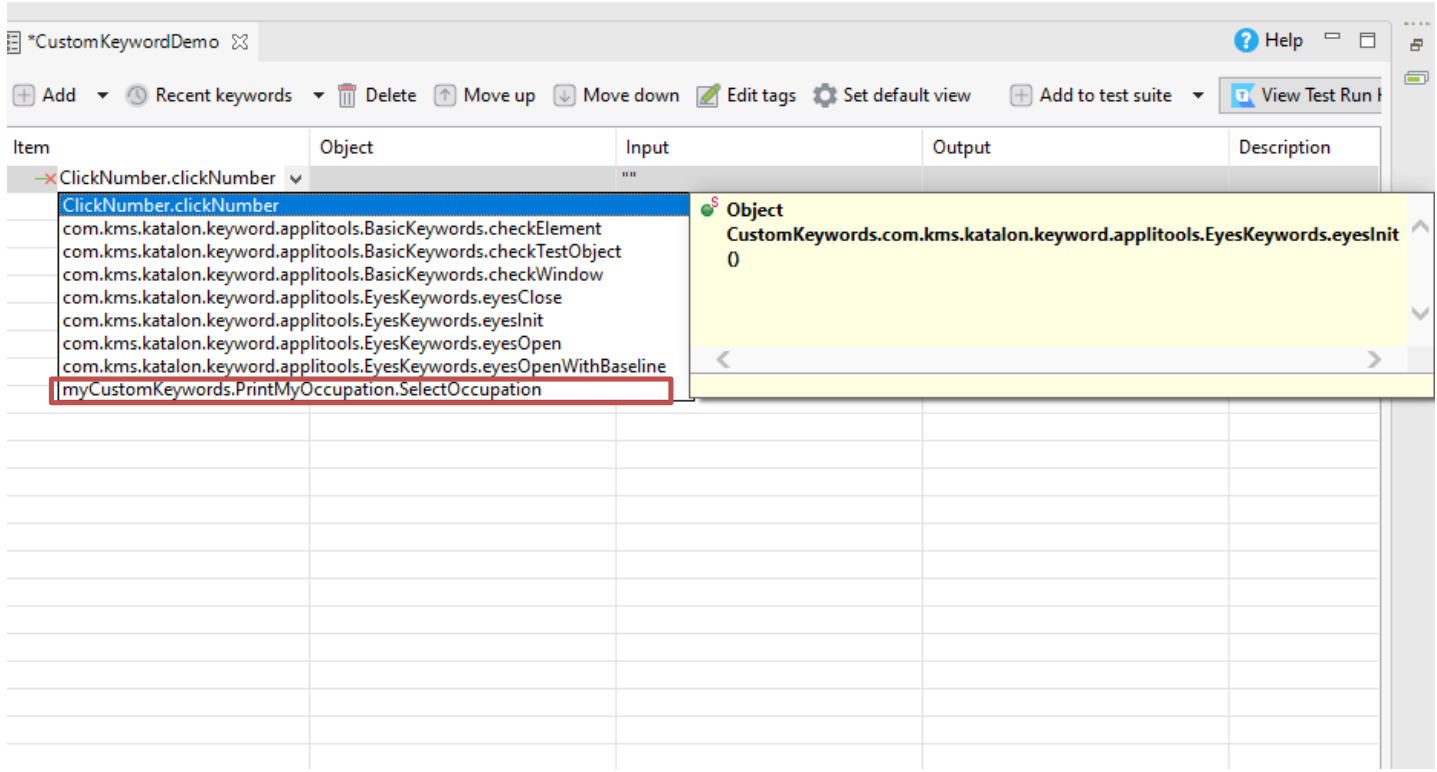
```groovy
public class PrintMyOccupation {

    @Keyword (keywordObject = "PrintFunction")
    def SelectOccupation (int input) {

        switch (input) {

            case 1:
            println("Automation Tester");
            break;

            case 2:
            println("Manual Tester");
            break;

            case 3:
            println("Automation and Manual Tester");
            break;


        }
    }
```

2.4 Custom Keywords

Step 4: Add Custom Keyword to Test Case

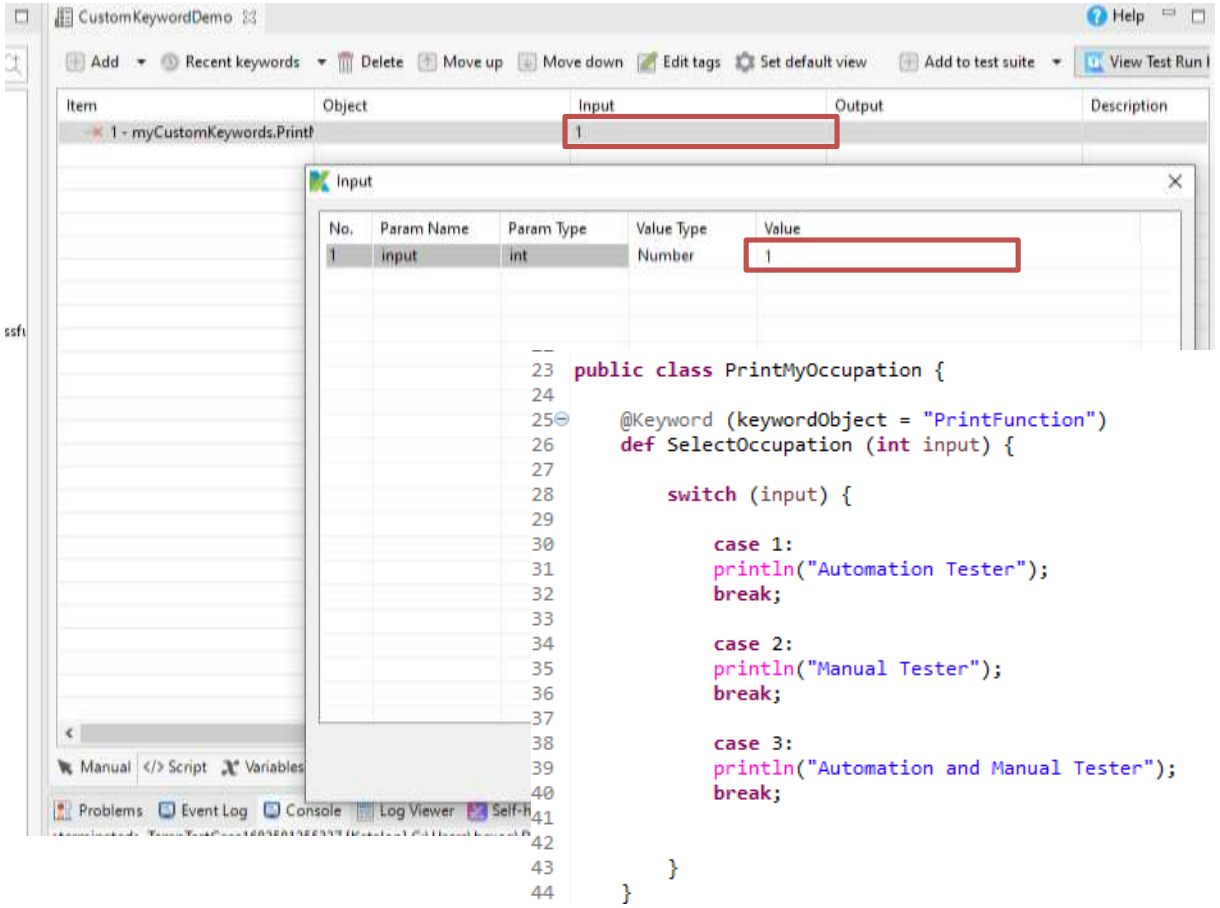
-Create a new test case named “CustomKeywordDemo” and “add” Custom Keyword and choose the function “SelectOccupation”





2.4 Custom Keywords

Step 5: Execute the Test Case



Change the Input to 1,2 or 3 and execute the test case while observing the result on the console.

Input 1 = `Automation Tester`

Input 2 = `Manual Tester`

Input 3 = `Automation and Manual Tester`

Q&A

TESTBITS

THE END

+60388005219

https://mytestbits.com/

contactus@mytestbits.com

B-01-02, Tamarind Square,Persiaran Multimedia, Cyber 10,
63000 Cyberjaya, Selangor, Malaysia