



Katalon Studio For Automated Testing

Sesi 3



Exploration⁺ of Katalon Studio Main Features

Exploration Of Katalon Studio Main Features

Preview Assignment 1

Make Appointment

Facility Hongkong CURA Healthcare Center

☒ Apply for hospital readmission

Healthcare Program ☐ Medicare ☒ Medicaid ☐ None

Visit Date (Required) 30/04/2017

Comment This is a comment

[Book Appointment](#)

Appointment Confirmation

Please be informed that your appointment has been booked as following:

Facility	Hongkong CURA Healthcare Center
Apply for hospital readmission	Yes
Healthcare Program	Medicaid
Visit Date	30/04/2017
Comment	This is a comment

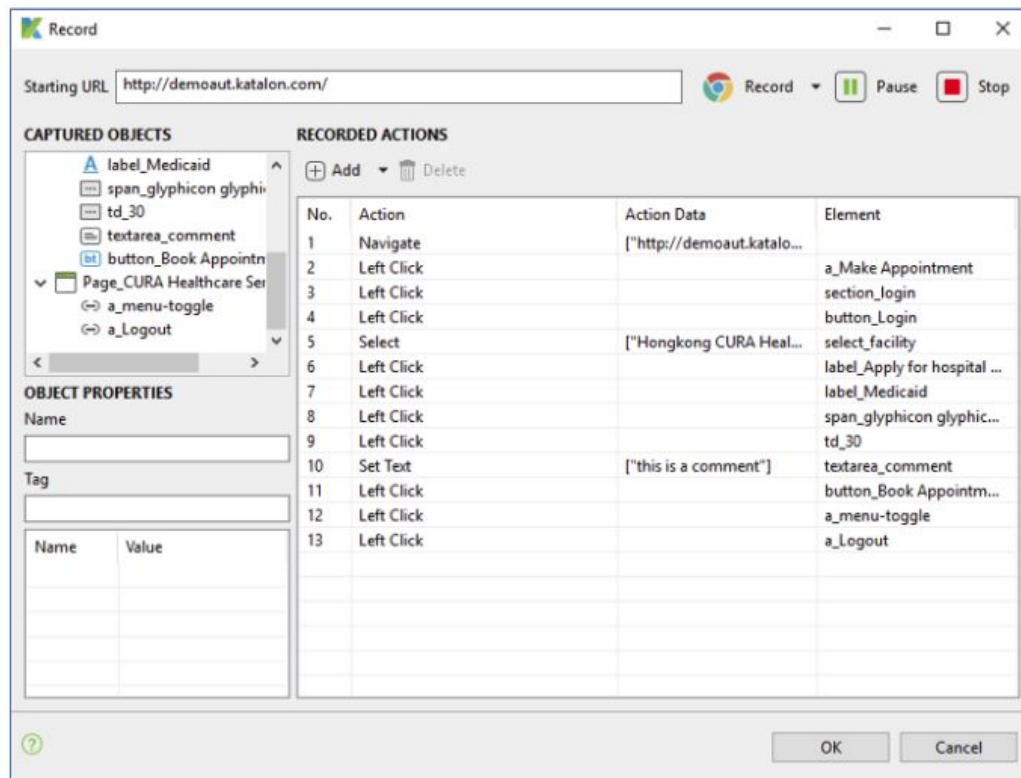
[Go to Homepage](#)

CURA Healthcare Service
Atlanta 550 Pharr Road NE Suite 525
Atlanta, GA 30305
(678) 813-1KMS
info@katalon.com

CURA Healthcare
[Home](#)
[History](#)
[Profile](#)
[Logout](#)

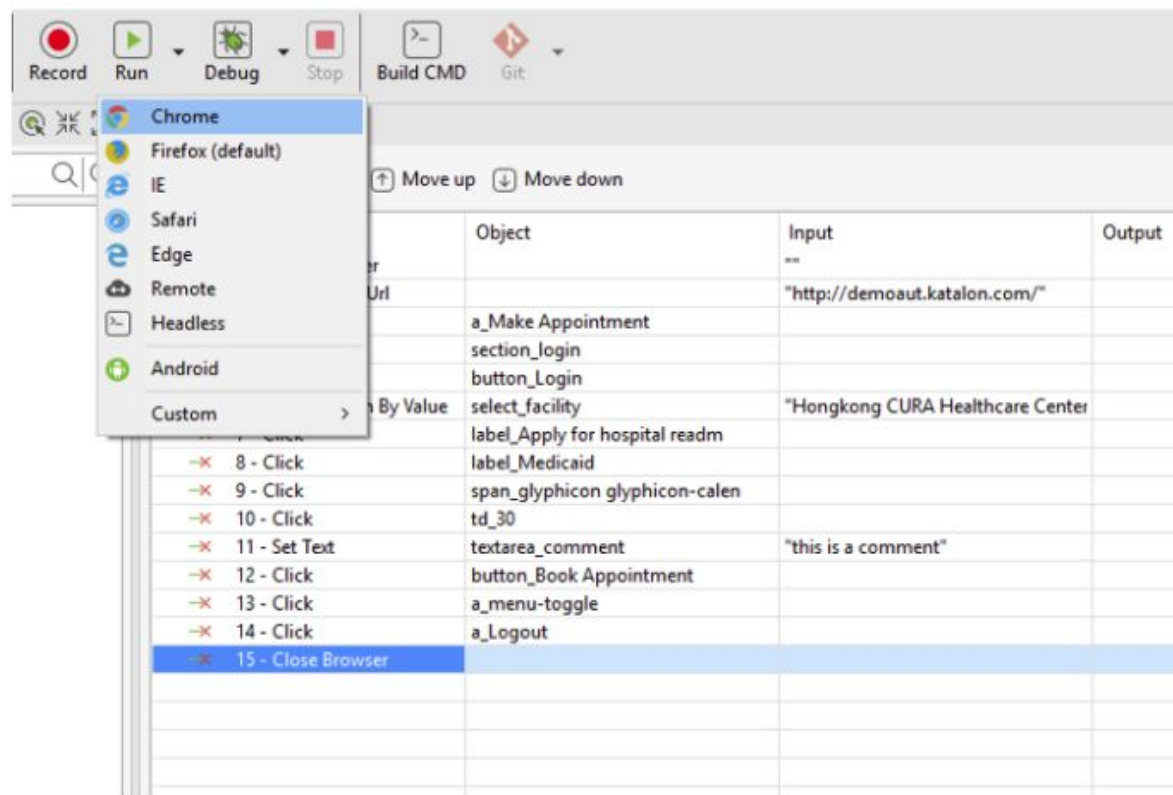


- On the Make Appointment page, fill in necessary appointment information and click on the Book Appointment button to book an appointment.
- The Appointment Confirmation page will show up. Click the Logout button to complete the recorded session.



Switch back to Katalon Studio Record dialog and you can see all the actions and web elements recorded.

Click on OK to finish the recording and generate necessary test steps and test scripts.



The screenshot shows the Katalon Studio Record dialog. On the left, there is a list of recorded actions, each with a red 'x' icon and a number. The actions are:

- 8 - Click
- 9 - Click
- 10 - Click
- 11 - Set Text
- 12 - Click
- 13 - Click
- 14 - Click
- 15 - Close Browser

On the right, there is a table with columns: Object, Input, and Output. The table contains the following data:

Object	Input	Output
Url	"http://demoaut.katalon.com/"	
a_Make Appointment		
section_login		
button_Login		
select_facility	"Hongkong CURA Healthcare Center"	
label_Apply for hospital readm		
label_Medicaid		
span_glyphicon glyphicon-calen		
td_30		
textarea_comment	"this is a comment"	
button_Book Appointment		
a_menu-toggle		
a_Logout		

The recorded test scripts should run smoothly, but you will notice the test fails at Step 14 (Click on “a_Logout” element) with the error message “Unable to click on object ‘Object Repository/Page CURA Healthcare Service (3)/a Logout””.

Don't worry, this can be fixed using the following steps.

Basic			
+ Add - Delete ↑ Move up ↓ Move down			
Item	Object	Input	Output
✖ 1 - Open Browser		""	
✖ 2 - Navigate To Url		"http://demoaut.katalon.com/"	
✖ 3 - Click	a_Make Appointment		
✖ 4 - Click	section_login		
✖ 5 - Click	button_Login		
✖ 6 - Select Option By Value	select_facility	"Hongkong CURA Healthcare Center"	
✖ 7 - Click	label_Apply for hospital readm		
✖ 8 - Click	label_Medicaid		
✖ 9 - Click	span_glyphicon glyphicon-calen		
✖ 10 - Click	td_30		
✖ 11 - Set Text	textarea_comment	"this is a comment"	
✖ 12 - Click	button_Book Appointment		
✖ 13 - Click	a_menu-toggle		
✖ 14 - Wait For Element Clickable	a_Logout	5	
✖ 15 - Click	a_Logout		
✖ 16 - Close Browser			



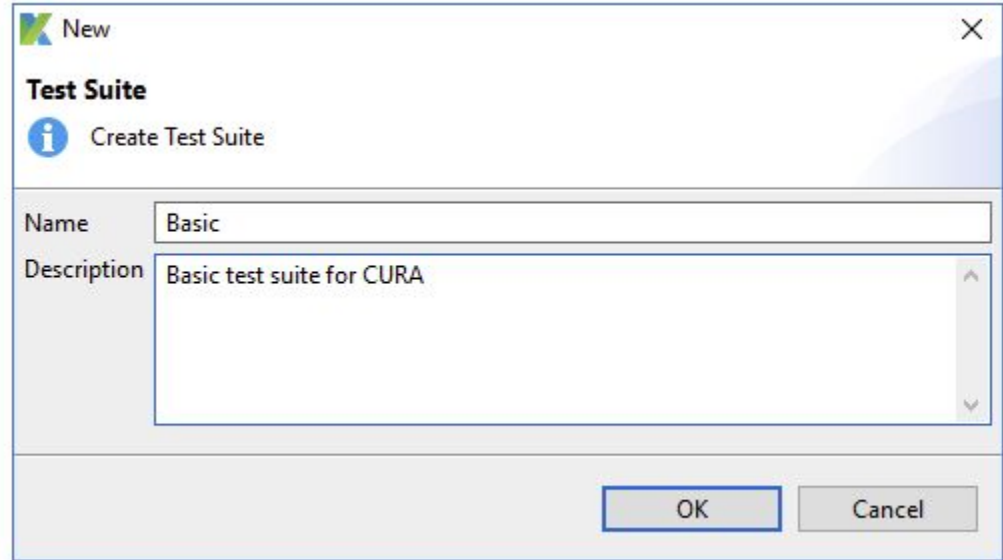
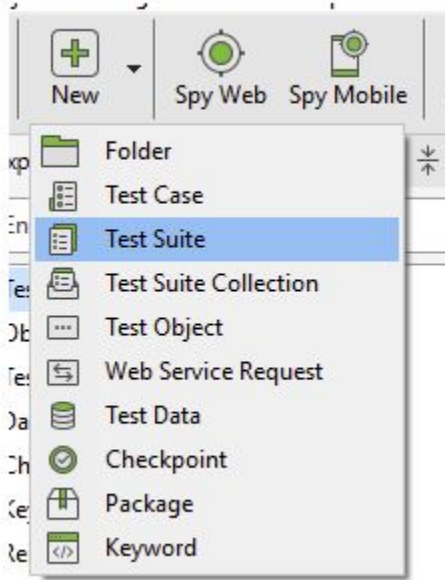
If you investigate to the failed message of the test execution, it includes the message “org.openqa.selenium.WebDriverException: unknown error: Element is not clickable at point”.

It happens because when the Side Menu of CURA website is open, the animation causes the Logout element to appear slowly and so Katalon Studio’s playback is unable to recognize it.

To fix this issue, we need to include a Wait step right before the logout step using the keyword `waitForElementClickable`, use “a_Logout” element as the object and change the timeout input to 5 seconds. Run the test case again, you will see the Log out button is found and the step passes.

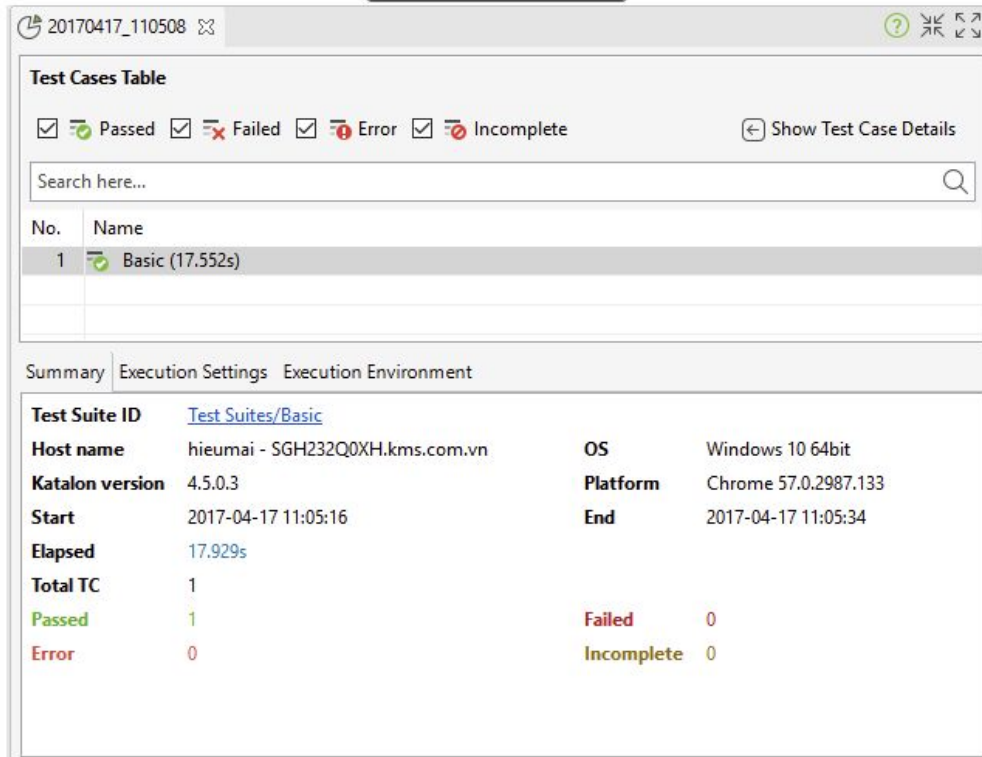
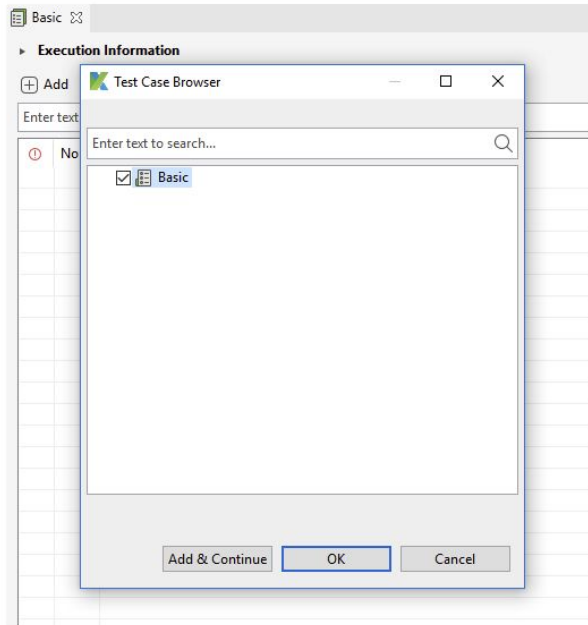
Exploration Of Katalon Studio Main Features

Generate Reports



To generate test reports, we need to have a test suite. A test suite in Katalon Studio is where you group test cases to run them together.

- To create a test suite, click on the New toolbar button and select the New Test Suite item.
- In the New Test Suite dialog, enter the required name and an optional description about your test suite, click OK to create a new test suite.



From the test suite UI, click the Add button and check the “Basic” test case then click OK to add the selected test case to the test suite.

Execute this test suite as we did with the test case by clicking on the Execution toolbar.

After the execution, you will notice that a new report folder is generated. In this folder, each of the child items represents an execution with the name indicating the starting time of the execution. Click on the first report item to view. The report is shown as above

20170417_110508 [1]

Test Cases Table

☒ Passed
 ☒ Failed
 ☒ Error
 ☒ Incomplete
 ☒ Hide Test Case Details

Search here...

No.	Name
1	Basic (173554)

Summary | Execution Settings | Execution Environment

Test Suite ID [Test Suite/Basic](#)
Host name hieumai - SGH232QD04.kms.com.vn **OS** Windows 10 64bit
Katalon version 4.5.0.3 **Platform** Chrome 57.0.2987.133
Start 2017-04-17 11:05:16 **End** 2017-04-17 11:05:34
Elapsed 17.929s
Total TC 1
Passed 1 **Failed** 0
Error 0 **Incomplete** 0

Test Case's Log

Test Log | Information | Integration

☒ Info
 ☒ Passed
 ☒ Failed
 ☒ Error
 ☒ Incomplete
 ☒ Warning
 ☒ Not Run

Search here...

Item	Description	Elapsed
> 1. openBrowser		5.577s
> 2. navigateToUrl		4.032s
> 3. click		0.531s
> 4. click		0.113s
> 5. click		0.726s
> 6. selectOptionByValue		0.257s
> 7. click		0.140s
> 8. click		0.106s
> 9. click		0.121s
> 10. click		0.104s
> 11. setText		0.193s
> 12. click		0.462s
> 13. click		0.102s
> 14. waitForElementClickable		0.109s
> 15. click		0.977s
> 16. closeBrowser		3.818s

Information | Image

Name waitForElementClickable
Start 2017-04-17 11:05:29 **End** 2017-04-17 11:05:29 **Elapsed** 0.109s
Description
Message Object 'Object Repository/Page_CURA Healthcare Service (3)/x_Login' is clickable

The test report's detail is shown as below (with “Show Test Case Details” selected)

Exploration Of Katalon Studio Main Features

Conclusion

Perlu diperhatikan : Dengan melakukan pengujian seperti yang sudah dilakukan dengan katalon :
record - stop - playback - generate report

Cukup mudah bukan ?

Lalu bagaimana jika kita ingin melakukan manual test dengan katalon?

Kita perlu tau mode script dan bagaimana cara melakukan pengujian dengan mode script pada GUI Katalon.



SCRIPT MODE⁺

Introductions

Also, user can record the test script or create test script in manual mode and then convert it into script mode by just clicking the Script tab. As stated in our previous tutorials, Katalon studio inherently supports Keyword-driven framework. It provides inbuilt keywords for WebUI, mobile app and web service automation.

Browser related keywords

`openBrowser()`

Opens browser with the URL provided, if no URL is provided then it will just open a blank browser.

`WebUI.openBrowser('')`

`closeBrowser()`

Closes a browser.

1 | `WebUI.closeBrowser()`

`navigateToUrl()`

Navigates to a url provided in the **Input** cell.

1 | `WebUI.navigateToUrl('https://artoftesting.com')`

`deleteAllCookies()`

Deletes browser cookies.

1 | `WebUI.deleteAllCookies()`

Let's first see some commonly used keywords provided by Katalon Studio for automating web applications.



HACKTIV8

Browser related keywords

`waitForPageLoad()`

Waits for page to load with the timeout value in seconds.

```
1 | WebUI.waitForPageLoad(5)
```

`refresh()`

Refreshes the browser.

```
1 | WebUI.refresh()
```

`back()`

Goes back to previous URL in the browser history.

```
1 | WebUI.back()
```

`forward()`

Goes to next URL in the browser history.

```
1 | WebUI.forward()
```



HACKTIV8

Web element related keywords

click()

Simulates mouse left-click on a web element located by the findTestObject method.

```
1 | WebUI.click(findTestObject('Page_home/btn_submit'))
```

doubleClick()

Simulates mouse double-click on a web element.

```
1 | WebUI.doubleClick(findTestObject('Page_home/btn_submit'))
```

rightClick()

Right clicks on a web element.

```
1 | WebUI.rightClick(findTestObject('Page_home/btn_submit'))
```

check()

Checks a check-box or radio button.

```
1 | WebUI.check(findTestObject('Page_home/chk_box1'))
```



HACKTIV8

Web element related keywords

`unchecked()`

Unchecks a check-box or radio button.

```
WebUI.unchecked(findTestObject('Page_home/chk_box1'))
```

`setText()`

Writes a text on a textbox or any input field (similar to `sendKeys()` in selenium)

```
1 | WebUI.setText(findTestObject('Page_login/txt_username'), username)
```

`selectOptionByValue()`

Selects a dropdown option by its value.

```
WebUI.selectOptionByValue(findTestObject('Page_login/gender'), 'Male', false)
```

`selectOptionByLabel()`

Selects a dropdown option by its visible text.

```
WebUI.selectOptionByLabel(findTestObject('Page_login/gender'), 'M', false)
```

`selectOptionByIndex()`

Selects a dropdown option by its index value.

```
WebUI.selectOptionByIndex(findTestObject('Page_login/gender'), 1)
```



Common wait related keywords

`waitForElementClickable()`

Waits for an element to be clickable with timeout value in seconds.

```
WebUI.waitForElementClickable(findTestObject('Page_home/btn_1'), 10)
```

`waitForElementVisible()`

Waits for an element to be visible with timeout value in seconds.

```
WebUI.waitForElementVisible(findTestObject('Page_home/btn_2'), 10)
```

`waitForElementAttributeValue()`

Waits till an element is not visible.

```
1 | WebUI.waitForElementAttributeValue(findTestObject('Page_home/btn_3')
```



Commonly used validation related keywords

`verifyElementPresent()`

Returns a boolean value of true/false based on the presence of an element.

```
WebUI.verifyElementPresent(findTestObject('Page_home/btn_1'), 20)
```

`verifyElementNotPresent()`

Returns a boolean value of true/false based on the absence of an element.

```
WebUI.verifyElementNotPresent(findTestObject('Page_home/btn_1'), 20)
```

`verifyElementText()`

Return true if the element's text is same as expected text else false.

```
1 | WebUI.verifyElementText(findTestObject('Page_home/btn_1'), 'Submit')
```

`verifyEqual()`

Return true if the actual value is same as expected value.

```
WebUI.verifyEqual(var1, 20)
```

`verifyNotEqual()`

Return false if the actual value is same as expected value.

```
WebUI.verifyNotEqual(var2, 20)
```



Other commonly used keywords

`acceptAlert()`

Simulates user action of accepting an alert or clicking 'Ok' on an alert dialog box.

`WebUI.acceptAlert()`

`dismissAlert()`

Simulates user action of dismissing an alert or clicking 'Cancel' on an alert dialog box.

```
1 | WebUI.dismissAlert()
```

`getAttribute()`

Used to get a particular attribute of a web element e.g. name, type etc.

`WebUI.getAttribute(findTestObject('Page_home/btn_1'), 'class')`

`getText()`

Used to get the element's text.

```
1 | WebUI.getText(findTestObject('Success_msg'))
```

`dragAndDropToObject()`

Used to get the element's text.

`WebUI.dragAndDropToObject(findTestObject('Page1/sourceElement'), findTest`



HACKTIV8

Other commonly used keywords

`switchToFrame()`

Switches to a particular iframe with timeout value of 10 seconds.

```
WebUI.switchToFrame(findTestObject('iframe_bqx'), 10)
```

`switchToWindowTitle()`

Switches to a particular window with the given title.

```
WebUI.switchToWindowTitle('ArtOftesting demo')
```

`closeWindowTitle()`

Return false if the actual value is same as expected value.

```
WebUI.closeWindowTitle('ArtOftesting demo')
```

`switchToDefaultContent()`

Switches to the parent window, called after performing some function on an iframe or a different window/tab.

```
1 | WebUI.switchToDefaultContent()
```

`takeScreenshot()`

Takes screenshot of the browser when called.

```
WebUI.takeScreenshot('D:\\screenshot.jpg')
```



Other commonly used keywords

`executeJavaScript()`

Used to execute javascript command.

```
1 | WebUI.executeJavaScript("{javascript code}", null)
```

`uploadFile()`

Used to perform file upload operation on a file type element.

```
WebUI.uploadFile(findTestObject('fileUpload'), 'D:\\screenshot.png')
```



Script Mode

Introductions

Setelah melihat contoh script yang ada/berlaku di Katalon, kali ini kita akan coba langsung dengan study case : GoogleCalculator dimana scenario nya :

1. Launch Google.
2. Enter 2+2 in Google's search bar.
3. Press enter.
4. Verify the result of 2+2 as 4.

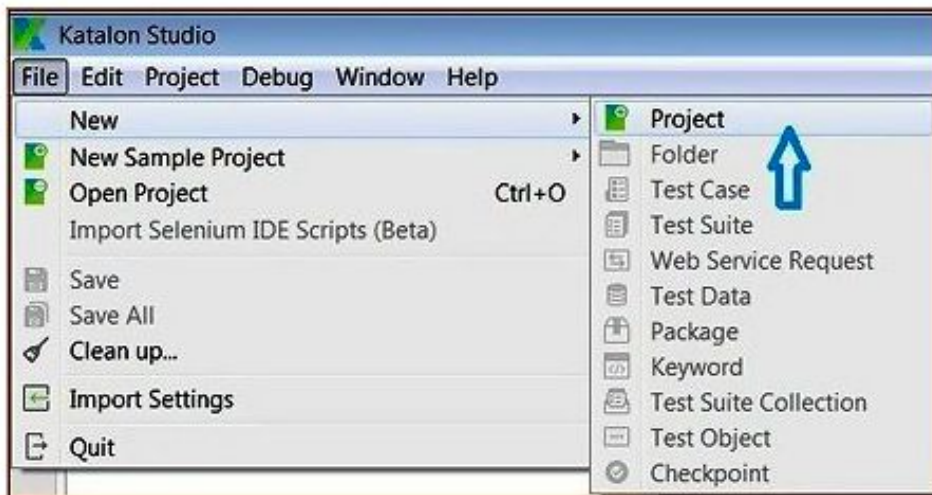


Script Mode

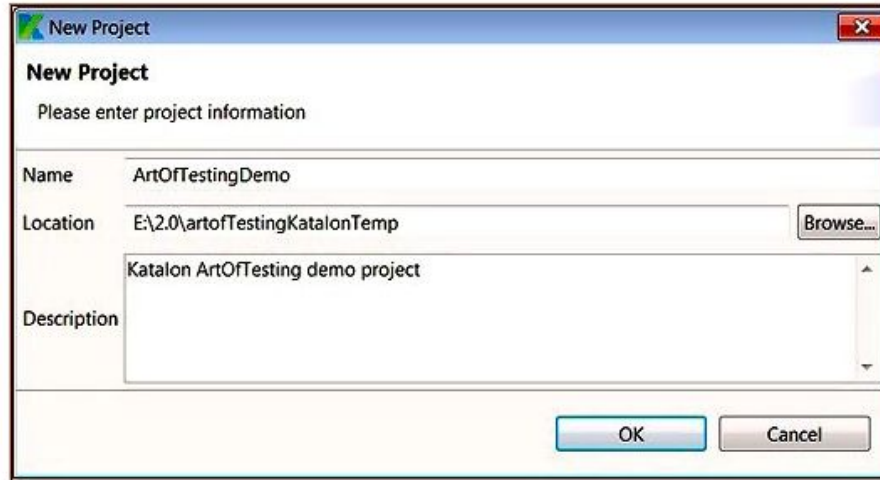
Case : GoogleCalculator

Let's first see, how to create a project in Katalon using record & playback feature.

1. On the Katalon Studio IDE, go to File menu and click on New->Project.



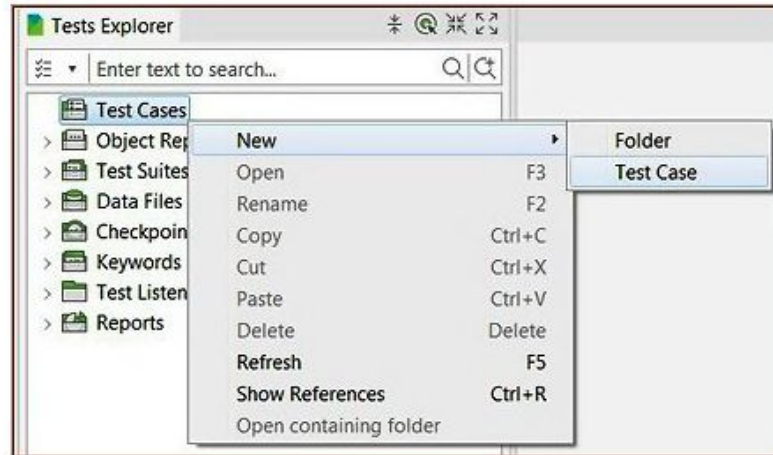
2. Name your Project, specify the project location, provide description(optional) and click OK.



The screenshot shows a 'New Project' dialog box with the following fields and content:

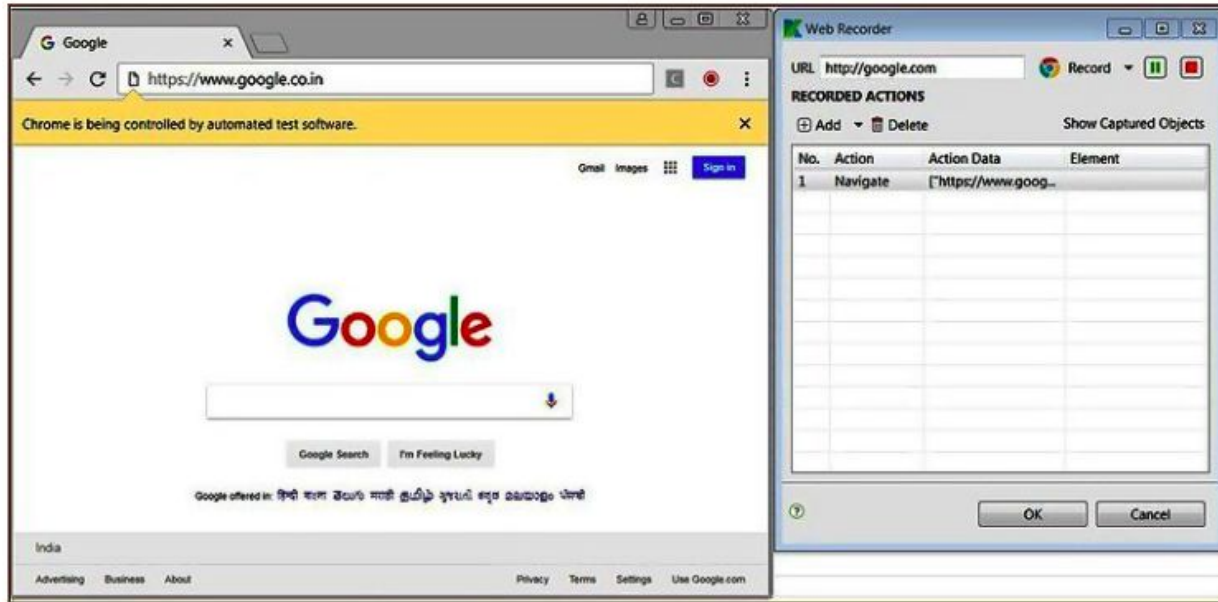
- Title Bar:** 'New Project' with a close button (X).
- Section Header:** 'New Project'.
- Instruction:** 'Please enter project information'.
- Name:** 'ArtOfTestingDemo'.
- Location:** 'E:\2.0\artofTestingKatalonTemp' with a 'Browse...' button.
- Description:** 'Katalon ArtOfTesting demo project'.
- Buttons:** 'OK' and 'Cancel' at the bottom right.

3. Next, we need to create a test case. For this right-click on 'Test Cases', click on **New** button on the toolbar and select **Test Case**.



Name your test case – 'GoogleCalc', provide a description and click OK.

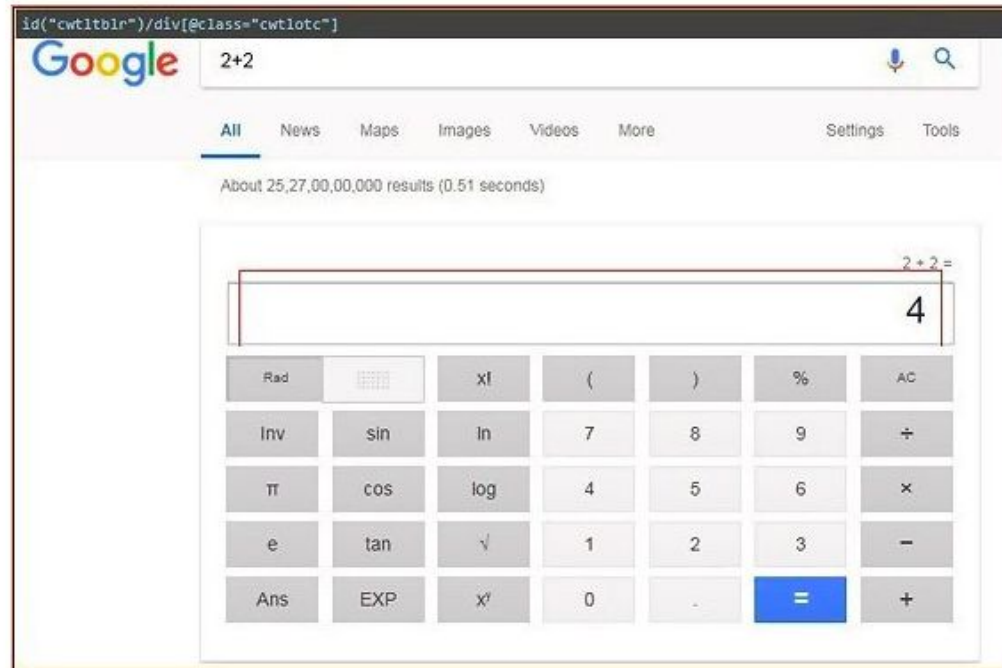
4. Now, we can record the script. For this, we need to click on **Record Web** button present in the toolbar. A new screen will appear, provide the URL e.g. “http://google.com” in the “Starting URL” field and select a browser. As soon as we select a browser, the corresponding browser will open with the URL. Any operation we perform on the browser will get recorded in Web Recorder.



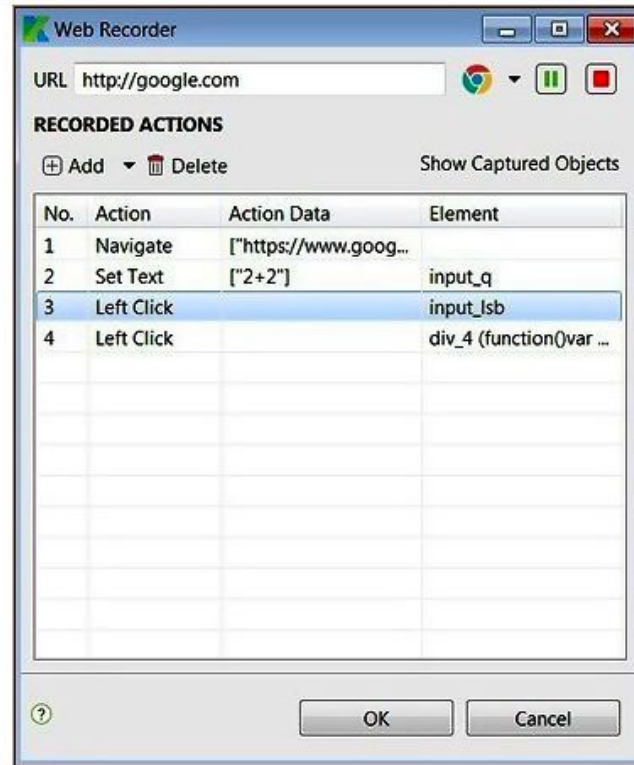
1. This concludes setting up a project for test case creation using Record and playback.
Now that we know, how to create a project on Katalon and start recording the scripts, let's focus on the Google Calculator test case creation.
- On the newly launched browser with google.com URL, click on the google search bar and enter 2+2. Keep an eye on the web recorder also, an action – “Set Text” will get recorded with “Action Data” as [“2+2”].



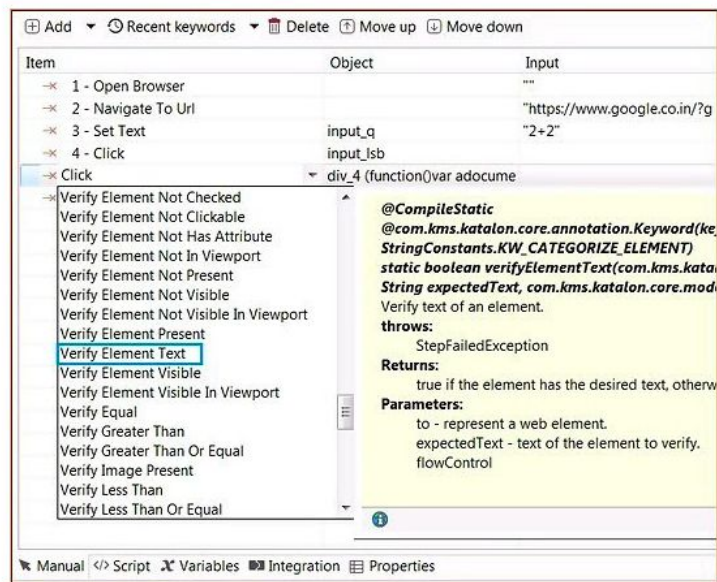
We will get redirected to the result page having the result of $2+2$ as '4'. Just click on the search bar having the result (Although clicking on the search bar is not required but we need to this to capture the element so that later on we can add assertion/validation action on that element).



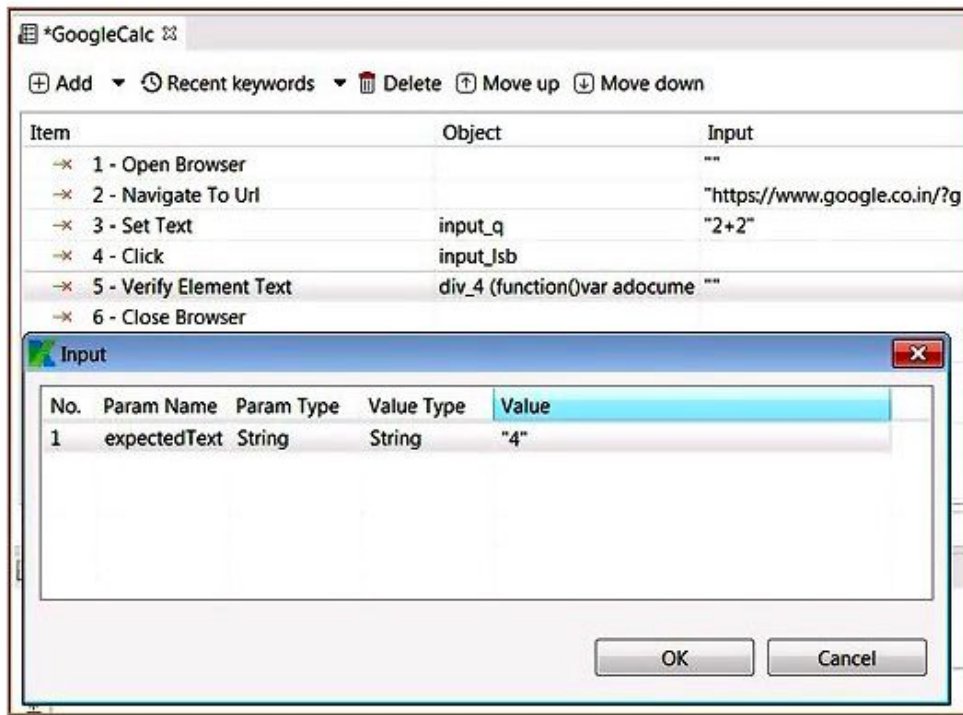
If everything is correct than your web recorder will look like this.



- Close the browser and click 'OK' on the web recorder. Now, you can see all the test steps recorded in the test case pane with its name – 'GoogleCalc'. Click on the 'Run' button to run the test script.
- If everything passes, we just need to add an Assertion or validation point which checks that the result of 2+2 is getting correctly calculated as '4'. For this, we will update the action in which we just clicked on the google search bar having the result of the addition. Instead of 'Click' action make it "Validate element text" by selecting the value from the action dropdown.



Now, double click Input cell corresponding to “Verify element text” action and enter ‘4’ in the ‘Value’ cell.



Google calculator test in script mode

```
WebUI.openBrowser('')
```

```
WebUI.navigateToUrl('https://www.google.co.in/')
```

```
WebUI.setText(findTestObject('Page_Google/input_q'), '2+2')
```

```
WebUI.click(findTestObject('Page_Google/input_lsb'))
```

```
WebUI.verifyElementText(findTestObject('Page_22 - Google Search (1)/div_4  
(function()var adocument'), '4')
```

```
WebUI.closeBrowser()
```



This completes the test script creation part. Now, we can execute our test script by clicking 'Run' button.

You can see the test steps getting executed in the selected browser and the result of the test as passed. Also, you can change the expected value to '5' instead of the correct value '4' and run the test to see it getting failed.



Variabel

+

Variabel

Introductions

Untuk Memperdalam pemahaman terkait script mode pada Katalon, pada penutup kali ini kita sekalian akan mengenal dengan Variabel.

Ada 2 jenis Variabel pada Katalon Studio yang akan selalu kita gunakan antara lain :

- A. Test Case Variabel
- B. Global Variabel

Script Mode

A. Test Case Variabel

Mengijinkan kita untuk pass data yang dinamis ke dalam test case.

Kita bisa membuat variabel didalam test case dan memanggil test case dengan variabel baik itu di manual ataupun script.

Pada Part mana yang dikatakan Test Case Variabel ?
Kita buka lagi contoh testing login pada CURA Test



Katalon Studio - Sample Web UI - [Location: E:\K\Sample Web UI]

File Edit Project Debug Window Help

Tests Explorer

Enter test to search...

- Test Cases
 - Blocks
 - Scenarios
 - 01 Verify Successful Login (Variables)
 - 02 Verify Successful Login (Profiles - G
 - 03 Verify Successful Login (Blocks)
 - 04 Verify Successful Appointment
- Unit Test Keywords
- Object Repository
- Test Suites
- Data Files
- Checkpoints
- Keywords
- Test Listeners
- Reports
- Profiles

01 Verify Successful Login (Variables)

Item	Object	Input	Output
1 - Comment		"Story: Login to CURA system"	
2 - Comment		"Given that the user has the valid login information"	
3 - Open Browser		"http://demoaut.katalon.com"	
4 - Click	makeAppointment		
5 - Set Text	username	username	
6 - Set Text	password	password	
7 - Comment		"When he logins to CURA system"	
8 - Click	login		
9 - Comment		"Then he should be able to login successfully"	
10 - Verify Element Present	appointment	5	landingPage
11 - Method Call Statement		println(landingPage)	
12 - Delay		2	
13 - Click	menuBar		
14 - Click	logout		
15 - Close Browser			

Manual Script Variables Integration Properties

Demo Login

Add Recent keywords Delete Move up Move down

Item	Object	Input
1 - Open Browser		...
2 - Navigate To Url		"http://demoaut.katalon.com/"
3 - Click	a_Make Appointment	
4 - Set Text	input_username	"John Doe"
5 - Set Text	input_password	"ThisIsNotAPassword"
6 - Click	button_Login	
7 - Click	i_fa_fa-bars	
8 - Click	a_Logout	
9 - Close Browser		



Katalon Studio - Sample Web UI - [Location: E:\K\Sample Web UI]

File Edit Project Debug Window Help

Tests Explorer

Enter text to search...

- Test Cases
 - Blocks
 - Scenarios
 - 01 Verify Successful Login (Variables)
 - 02 Verify Successful Login (Profiles - G
 - 03 Verify Successful Login (Blocks)
 - 04 Verify Successful Appointment
 - Unit Test Keywords
- Object Repository
- Test Suites
- Data Files
- Checkpoints
- Keywords
- Test Listeners
- Reports
- Profiles

01 Verify Successful Login (Variables)

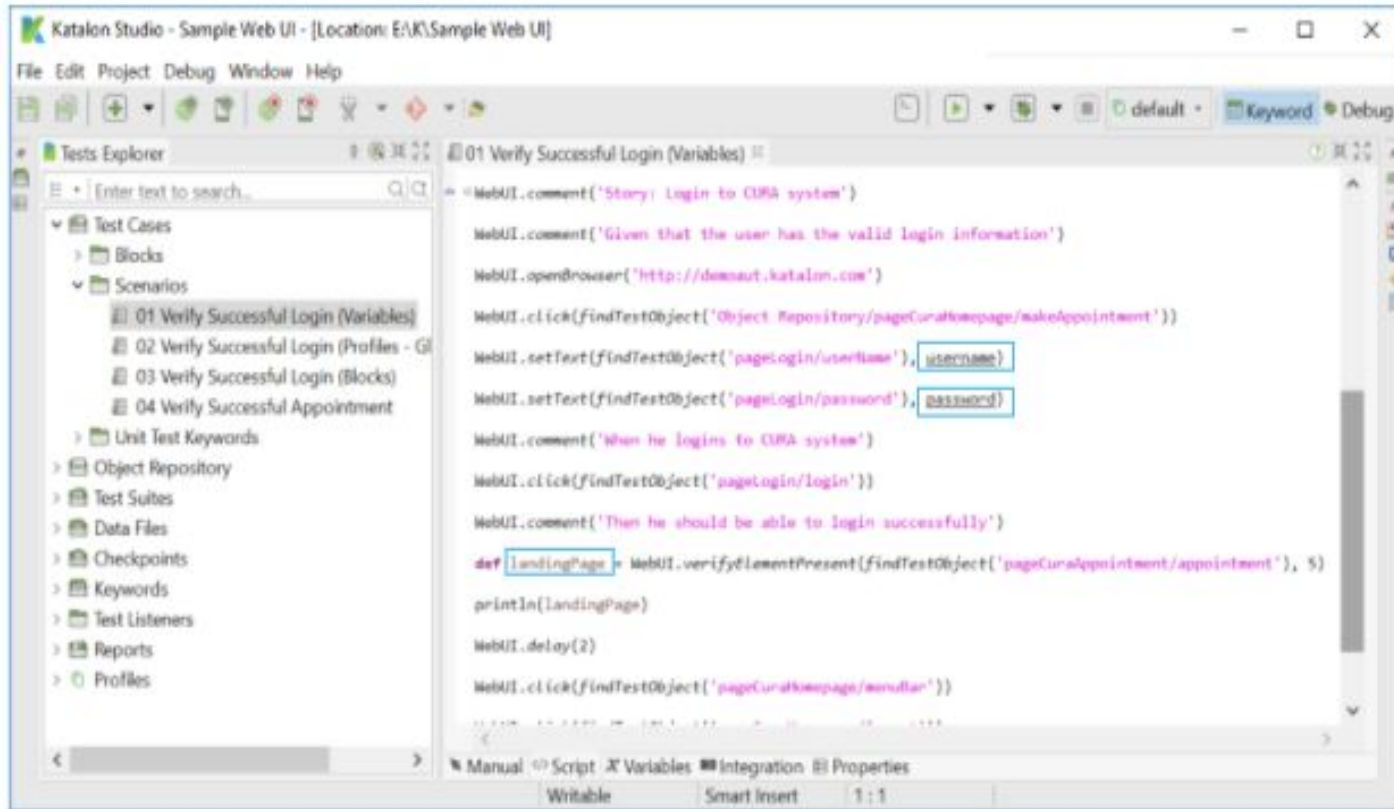
Add Delete Clear Move up Move down

No.	Name	Default value type	Default value
1	username	String	"John Doe"
2	password	String	"ThisIsNotAPassword"

Manual Script Variables Integration Properties



Lalu bagaimana dan bentuknya variabel pada mode script ?



The screenshot shows the Katalon Studio interface with a test script titled "01 Verify Successful Login (Variables)". The script is written in a keyword-driven format using WebUI methods. The left sidebar shows the project structure with "Test Cases" and "Scenarios" folders. The main editor displays the following code:

```
WebUI.comment('Story: Login to CURA system')
WebUI.comment('Given that the user has the valid login information')
WebUI.openBrowser('http://demoaut.katalon.com')
WebUI.click(findTestObject('Object Repository/pageCuraHomepage/makeAppointment'))
WebUI.setText(findTestObject('pageLogin/username'), username)
WebUI.setText(findTestObject('pageLogin/password'), password)
WebUI.comment('When he logs in to CURA system')
WebUI.click(findTestObject('pageLogin/login'))
WebUI.comment('Then he should be able to login successfully')
def landingPage = WebUI.verifyElementPresent(findTestObject('pageCuraAppointment/appointment'), 5)
println(landingPage)
WebUI.delay(2)
WebUI.click(findTestObject('pageCuraHomepage/menuBar'))
```

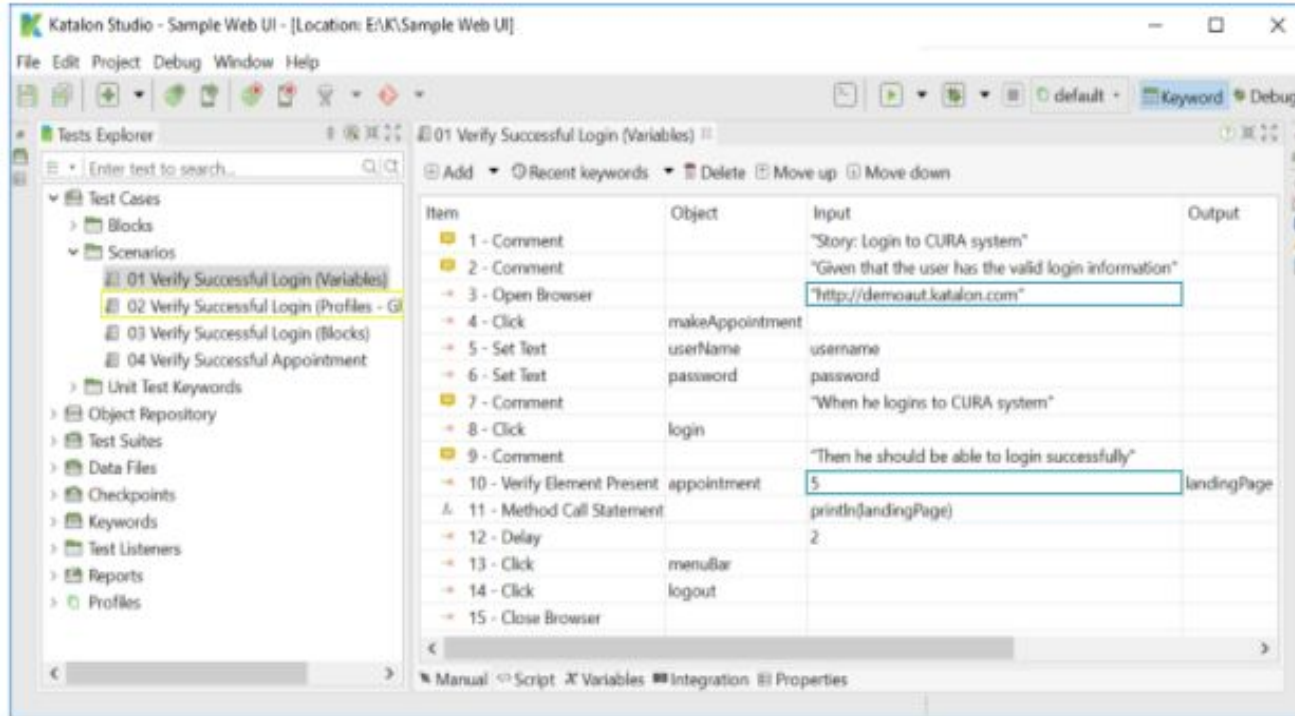
The variables `username` and `password` are used in the `setText` method calls. The variable `landingPage` is defined and used in the `verifyElementPresent` and `println` methods. The bottom status bar shows "Manual", "Script", "Variables", "Integration", and "Properties" tabs, with "Script" selected.



Script Mode

B. Global Variabel

A global variable is a variable defined in the execution profile and can be used in a test case, test object, web service object, and email configuration in a project.



Katalon Studio - Sample Web UI - [Location: E:\K\Sample Web UI]

File Edit Project Debug Window Help

Tests Explorer

Enter text to search...

- Test Cases
 - Blocks
 - Scenarios
 - 01 Verify Successful Login (Variables)
 - 02 Verify Successful Login (Profiles - Global Variables)
 - 03 Verify Successful Login (Blocks)
 - 04 Verify Successful Appointment
- Unit Test Keywords
- Object Repository
- Test Suites
- Data Files
- Checkpoints
- Keywords
- Test Listeners
- Reports
- Profiles
 - default
 - demoAUT

02 Verify Successful Login (Profiles - Global Variables)

Add Recent keywords Delete Move up Move down

Item	Object	Input	Output	Description
1 - Comment		"Story: Login to CURA sy		
2 - Comment		"Given that the user has		
3 - Open Browser		GlobalVariable.siteUrl		
4 - Set View Port S		1366; 1024		iPad Pro sized b
5 - Click	makeAppointment			
6 - Set Test	username	username		
7 - Set Test	password	password		
8 - Comment		"When he logs in to CUR		When he logs in
9 - Click	login			
10 - Comment		"Then he should be able		
11 - Verify Element	appointment	GlobalVariable.shortTime	landingPage	Then he should

Katalon Help 01 Verify Successful Log... 02 Verify Successful Log... demoAUT

Add Edit Delete Clear

Name	Value	Description
timeout	10	
siteUrl	'http://demoaut.katalon.com'	
shortTimeout	5	



Tests Explorer

Enter text to search...

- Test Cases
 - Blocks
 - BookAppointment
 - Login
 - Logout
 - Open
 - Scenarios
 - Unit Test Keywords
- Object Repository
 - pageAppointmentConfirmation
 - pageCuraAppointment
 - appointment
 - bookAppointment
 - comment
 - facility
 - medicaid
 - medicare
 - none
 - readmission

medicaid BookAppointment

Settings

Object's Properties

Selection Method ☒ Basic ☐ CSS ☐ XPath

+ Add - Delete Clear

Name	Condition	Value	<input checked="" type="checkbox"/> Detect object
id	equals	radio_program_medicaid	<input checked="" type="checkbox"/>
tag	equals	input	<input checked="" type="checkbox"/>
type	equals	radio	<input checked="" type="checkbox"/>

Selector Editor

```
//input[@id = 'radio_program_medicaid' and @type = 'radio']
```



Tests Explorer

Enter text to search...

- Test Cases
 - Blocks
 - BookAppointment
 - Login
 - Logout
 - Open
 - Scenarios
 - Unit Test Keywords
- Object Repository
 - pageAppointmentConfirmation
 - pageCuraAppointment
 - appointment
 - bookAppointment
 - comment
 - facility
 - medicaid
 - medicare
 - none
 - readmission

```
'if readmission then tick readmission'
if (readmission) {
    WebUI.check(findTestObject('pageCuraAppointment/readmission'))
}

switch (healthcareProgram) {
    case healthcareProgram = 'Medicare':
        'Case of Medicare then check medicare radio button'
        WebUI.check(findTestObject('pageCuraAppointment/medicare'))

        break
    case healthcareProgram = 'Medicaid':
        'Case of Medicare then check medicaid radio button'
        WebUI.check(findTestObject('pageCuraAppointment/medicaid'))

        break
}
```

Manual </> Script Variables Integration Properties

Problems Console Search Log Viewer

\QfindTestObject('Object Repository/pageCuraAppointment/medicaid')\E]
\QfindTestObject('pageCuraAppointment/medicaid')\E]\QfindTestObject('Object
Repository/pageCuraAppointment/medicaid')\E]
\QfindTestObject('pageCuraAppointment/medicaid')\E](<value>\QObject
Repository/pageCuraAppointment/medicaid\E</value>') - 1 match in 'settings', 'Checkpoints', ... (*.groovy, *.rs)

- BookAppointment
 - Script1524951660963
 - 37: WebUI.check(findTestObject('pageCuraAppointment/medicaid'))





Tests Explorer

Enter text to search...

- Test Cases
 - Blocks
 - BookAppointment
 - Login
 - Logout
 - Open
 - Scenarios
 - 01 Verify Successful Login (Variables)
 - 02 Verify Successful Login (Profiles - GL)
 - 03 Verify Successful Login (Blocks)
 - 04 Verify Successful Appointment
 - Unit Test Keywords
 - Object Repository
 - Test Suites
 - Data Files
 - Checkpoints
 - Keywords
 - Test Listeners
 - Reports

03 Verify Successful Login (Blocks)

Add Recent keywords Delete Move up Move down

Item	Object	Input	Output	Description
1 - Comment		"Story: Login to CURA system"		
2 - Call Test Case	Open			Given that the user has the valid login information
3 - Call Test Case	Login	"username":username, "password"		When he logs in to CURA system
4 - Verify Element Present	appointment	5	landingPage	Then he should be able to login successfully
5 - Call Test Case	Logout			
6 - Close Browser				

Test Case's Log

Test Log Information Integration

☒ Info ☒ Passed ☒ Failed ☒ Error ☒ Incomplete ☒ Warning

Search here...

Item	Description	Elapsed
1. comment		0.030s
2. callTestCase	Given that the user has the valid login information	7.423s
3. callTestCase	When he logs in to CURA system	1.884s
4. Statement - landi	Then he should be able to login successfully	0.019s
5. closeBrowser		0.862s



Katalon Studio - Sample Web UI - [Location: EMK\Sample Web UI]

File Edit Project Debug Window **Help**

Tests Explorer

- Test Cases
 - Blocks
 - BookAppointment
 - Login**
 - Logout
 - Open
 - Scenarios
 - 01 Verify Successful L
 - 02 Verify Successful L
 - 03 Verify Successful L
 - 04 Veriv Successful /

Help menu:

- About
- Check for Updates...
- Katalon Help
- Quick Guide
- Report Issue
- Encrypt Text**

Encrypt Text dialog:

Raw Text: ThisIsNotAPassword

Encrypted Text: g3/DOGG74jC3Flr3yH+3D/yKbOqqUNM

Buttons: Copy and Close, Cancel

Background interface details:

- Keyword: demoAUT
- Debug
- Table:

	Default value type	Default value
1	String	"John Doe"
2	String	"g3/DOGG74jC3Flr3yH+3D/yKbOqqUNM"
- Variables tab selected

