



Automated Testing with Katalon Studio

27–28 Apr. 2019

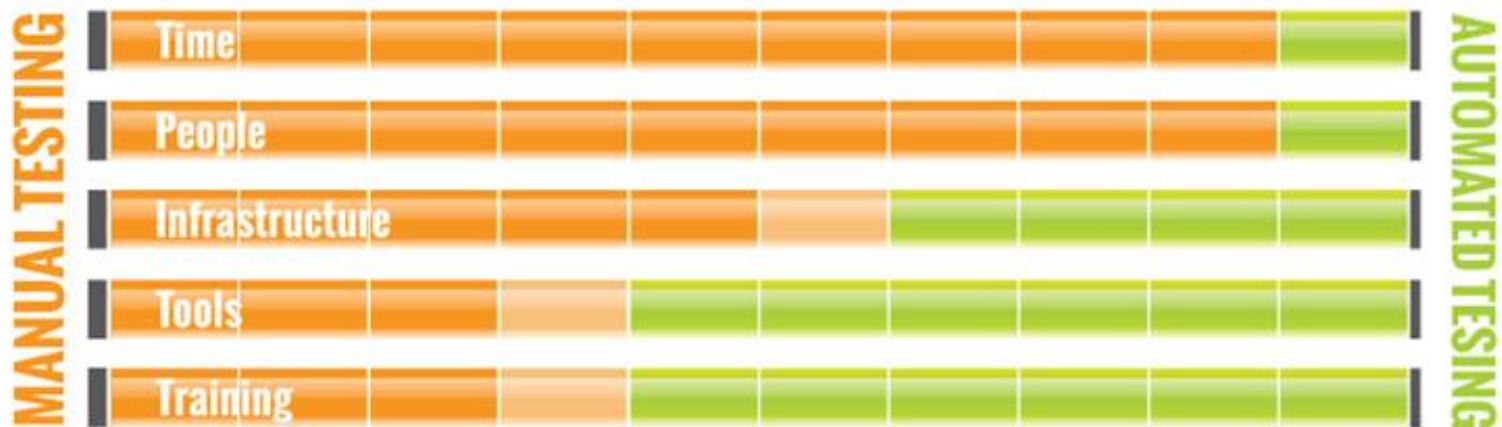
4–5 May 2019 @IT MJU

Assoc. Rangsit Sirirangsi

www.indythaitester.com

What is Automation Testing

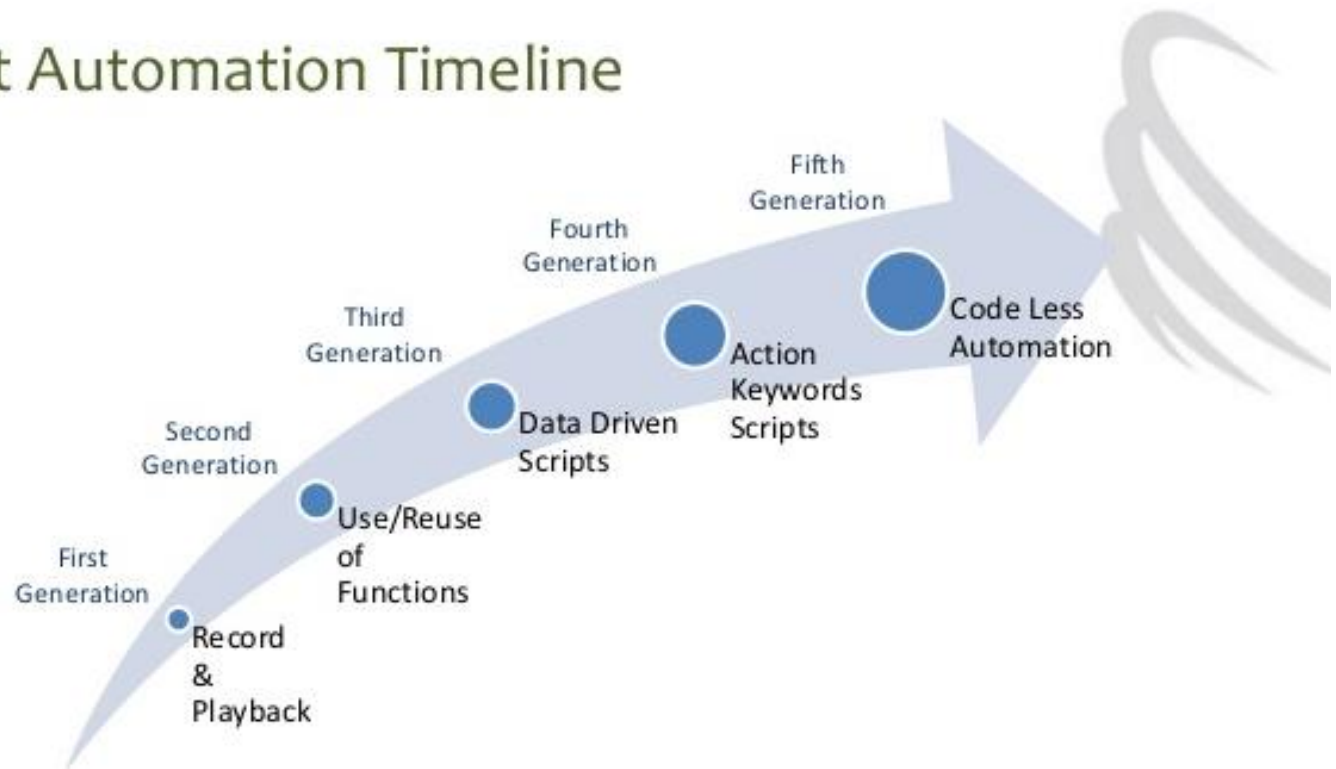
- **Test automation** เป็นการใช้ซอฟต์แวร์เพื่อควบคุมการประมวลผลการทดสอบ โดยเปรียบเทียบผลลัพธ์ที่เกิดขึ้นจริงกับผลลัพธ์ที่คาดหวัง
- ชนิดของการทดสอบที่สามารถทำงานแบบอัตโนมัติ
 - **Functional testing / Regression testing**
 - **Performance testing**



Automation Frameworks Evolution

- Automation Framework ที่ใช้ในการทดสอบปัจจุบันจะประกอบไปด้วยคุณสมบัติดังต่อไปนี้

Test Automation Timeline



Suggested : Automation tools

- **Open-source tools (coding experience required)**

- Selenium
- Appium
- Robot Framework
- Cucumber



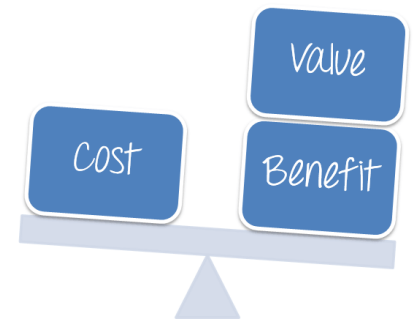
- **Free tools (limited community's support)**

- [Katalon Studio](#)

- **Commercial tools (high support & license cost)**

- [TestComplete](#)
- UFT
- [Ranorex](#)

Selection criteria for Automation tools



Best Automation Testing Tools for 2019

Product	 Selenium	 Katalon Studio	 Unified Functional Testing	 TestComplete	 SoapUI
Available since	2004	2015	1998	1999	2005
Application Under Test	Web apps	Web (UI & API), Mobile apps	Web (UI & API), Mobile, Desktop, Packaged apps	Web (UI & API), Mobile, Desktop apps	API/Web service
Pricing	Free	Free	\$\$\$\$	\$\$\$	\$\$
Supported Platforms	Windows Linux OS X	Windows Linux OS X	Windows	Windows	Windows Linux OS X
Scripting languages	Java, C#, Perl, Python, JavaScript, Ruby, PHP	Java/Groovy	VBScript	JavaScript, Python, VBScript, JScript, Delphi, C++ and C#	Groovy
Programming skills	Advanced skills needed to integrate various tools	Not required. Recommended for advanced test scripts	Not required. Recommended for advanced test scripts	Not required. Recommended for advanced test scripts	Not required. Recommended for advanced test scripts
Ease of Installation and Use	Require advanced skills to install and use	Easy to setup and use	Complex in installation. Need training to properly use the tool	Easy to setup. Need training to properly use the tool	Easy to setup and use

Katalon Studio

- เป็นเครื่องมือทดสอบอัตโนมัติที่ใช้เวลารวดเร็วในการเรียนรู้ สะดวกต่อการใช้งาน
- เป็นทางเลือกสำหรับเครื่องมือทดสอบอัตโนมัติที่มีค่าใช้จ่าย
- พัฒนาขึ้นโดยอาศัยโครงสร้างบางส่วนจาก Eclipse SDK ดังนั้นจึงสะดวกต่อการใช้งานสำหรับผู้ใช้ที่เคยใช้งานจาวามาก่อน
- Katalon Studio เริ่มต้นพัฒนาจาก KMS Technology R&D ในปี 2015
- KMS Technology เผยแพร่ Katalon Studio เครื่องมือทดสอบอัตโนมัติที่ไม่มีค่าใช้จ่าย และจัดตั้ง Katalon LLC. ใน January 2018

Katalon Studio

- โดยมีลักษณะเป็นเครื่องมือทดสอบอัตโนมัติที่มีประสิทธิภาพและใช้งานได้ง่าย
- ทำงานครอบงำบน Appium และ Selenium ดังนั้นจึงสามารถใช้ในการทดสอบเว็บแอปฯ โมบายแอปฯ และใช้ทดสอบ API ได้
- ใช้ Apache Groovy ซึ่งสามารถใช้งานร่วมกับภาษาจาวา ที่มีลักษณะคล้ายกับ JRuby และ Jython
- เป็นเครื่องมือที่ไม่มีค่าใช้จ่าย แต่ไม่ใช่ซอฟต์แวร์รหัสเปิด (open source)
- สามารถใช้งานร่วมกับ Git, Jenkins, qTest และ Jira

Advantages of Katalon Studio

- ในกรณีที่ทดสอบเว็บแอปฯ การติดตั้งใช้งานสามารถทำได้ง่าย โดยไม่ต้องโหลดส่วนประกอบอื่น ๆ เพิ่มเติม
- Katalon ถูกออกแบบมาสำหรับผู้ใช้ที่ไม่ใช่โปรแกรมเมอร์ โดยมีคุณสมบัติ Record & Playback สำหรับการสร้าง Test Case สคริปต์แบบ Manual ซึ่งจะช่วยให้ Manual tester สามารถสร้างการทดสอบอัตโนมัติได้โดยง่าย
- ส่วนผู้ใช้ที่เป็นโปรแกรมเมอร์สามารถเพิ่มเติมความสามารถของ Test Case ได้โดยการโปรแกรมลงใน Script Mode ภายใน Katalon ได้โดยตรง
- นอกจากนั้นยังมี Built-in templates และ frameworks ที่ช่วยลดเวลาในการสร้าง Test Case ส่งผลให้ผู้ใช้สามารถเน้นไปที่สคริปต์สำหรับการทดสอบอัตโนมัติได้ตามต้องการ

Advantages of Katalon Studio

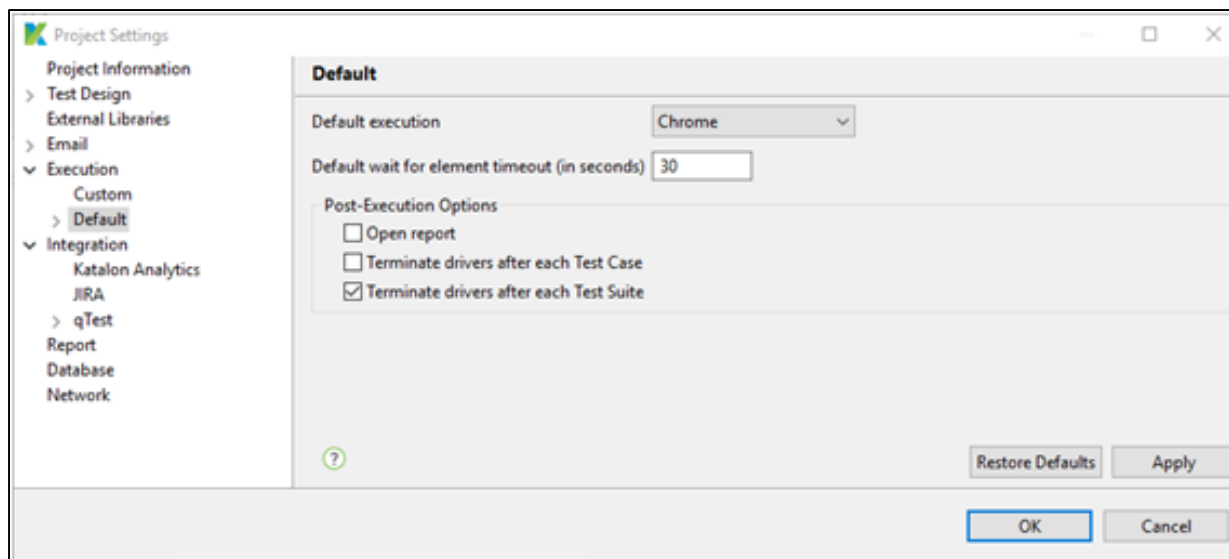
- สคริปต์ที่ถูกบันทึกไว้จากการทำงานของ browser หนึ่งสามารถนำไปรันกับ browser อื่น ๆ ส่งผลให้การทดสอบแบบ **cross browser** สามารถทำได้ง่าย และรวดเร็ว
- แสดงผลลัพธ์ของการทดสอบได้หลายรูปแบบ และสามารถ export เป็นไฟล์ pdf และ CSV ได้
- นอกจากนั้นยังมีกลไก logging และ screenshot เพื่อแจ้งให้ผู้ใช้ทราบ ในกรณีที่เกิดความล้มเหลวในการทดสอบ



Why Katalon Studio

- **Setup Project:**

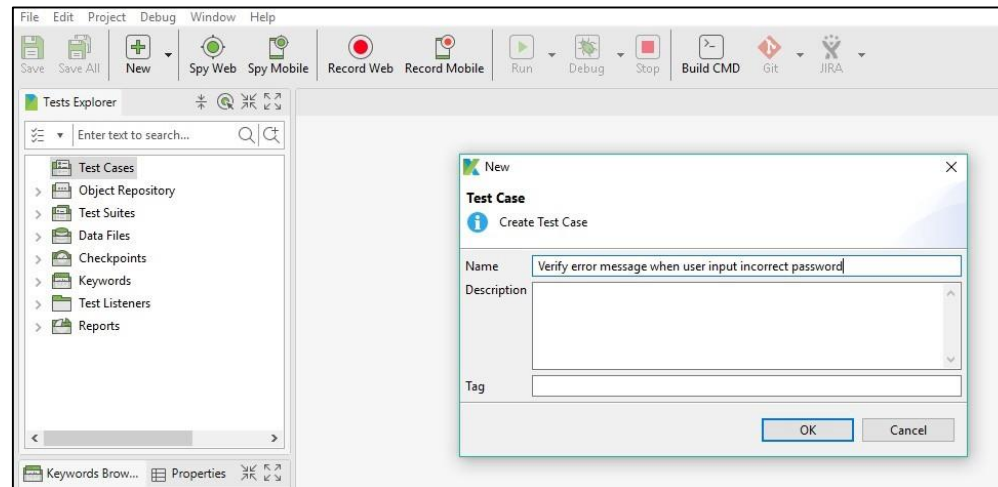
- มีรูปแบบการทำงานที่หลากหลายเพื่อให้ผู้ใช้สามารถเลือกจัดการ test cases, object repository และ keywords ได้ตามต้องการ
- สนับสนุนการทดสอบเว็บ โมบายทั้ง Android และ iOS รวมถึงการทดสอบ API สำหรับทุก ๆ ระบบปฏิบัติการ



Why Katalon Studio

- **Create Tests:**

- Records การกระทำของผู้ใช้และสร้างให้อยู่ในรูปของสคริปต์โดยใช้ built-in keywords
- ยอมให้ผู้ใช้สามารถสร้าง Keyword เพื่อใช้งานเพิ่มเติมได้ตามต้องการ
- กลไก recorder สามารถตรวจจับ object properties ได้อย่างถูกต้อง



Why Katalon Studio

- **Execute Tests:**

- รัน test cases หรือ test suites ได้โดยใช้ข้อมูลและค่าโครงสร้างแบบหลายชนิด
- ประกอบไปด้วย CI integration console ที่สามารถใช้ในการประมวลผลระยะไกล
- รันการทดสอบได้กับบราวเซอร์ชนิดต่าง ๆ

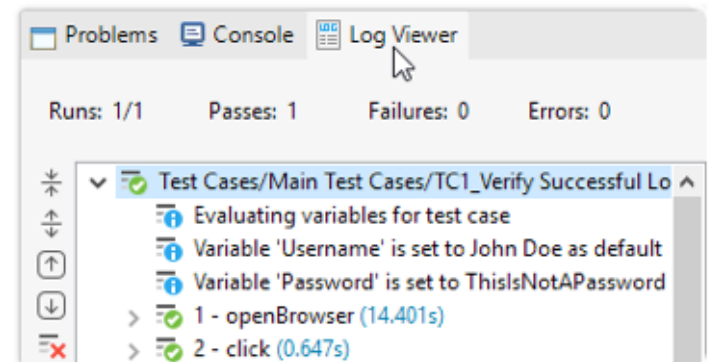
Step 1: Record to generate new Test Case



Step 2: Execute the generated Test Case



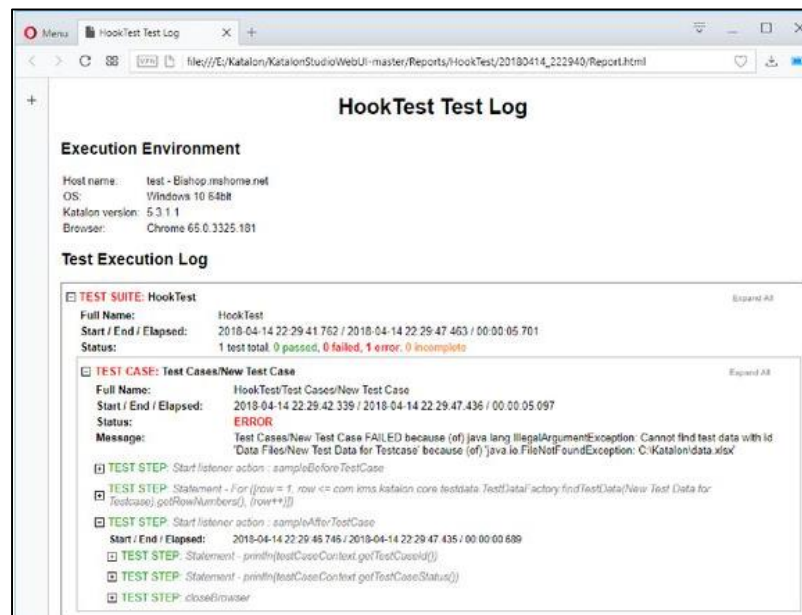
Step 3: View execution log



Why Katalon Studio

- **View Reports:**

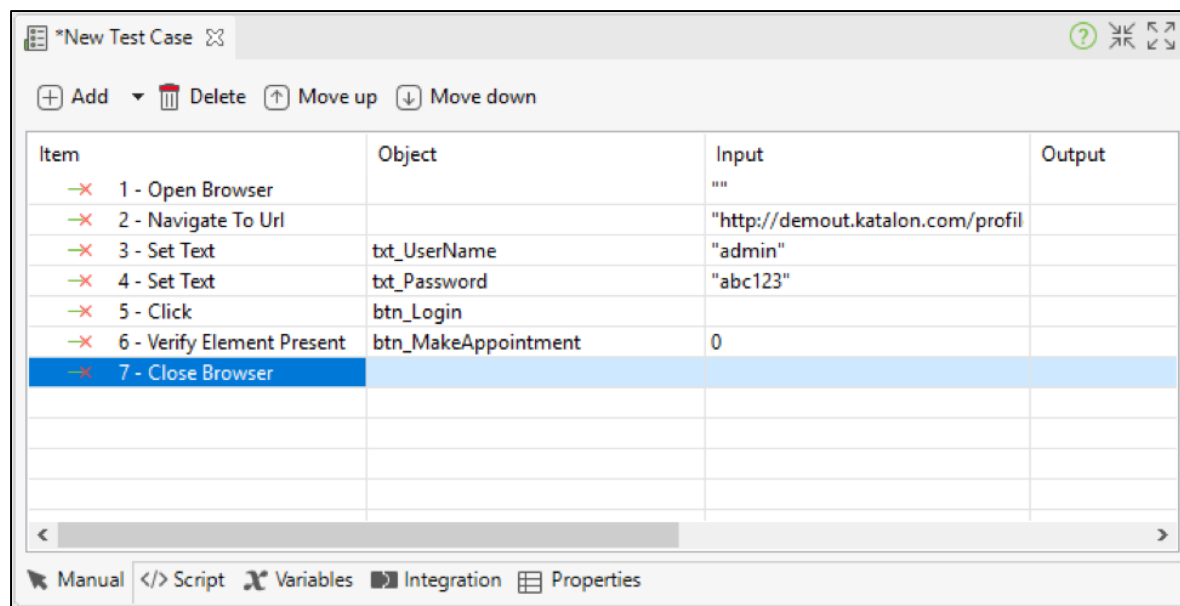
- ประกอบด้วย report ต่างๆ หลายแบบ รวมไปถึง log ไฟล์และscreenshots
- สามารถใช้ส่วนต่อขยายของ Selenium และ Appium logs เพื่อคุณสมบัติของการวิเคราะห์ได้มากขึ้น



Why Katalon Studio

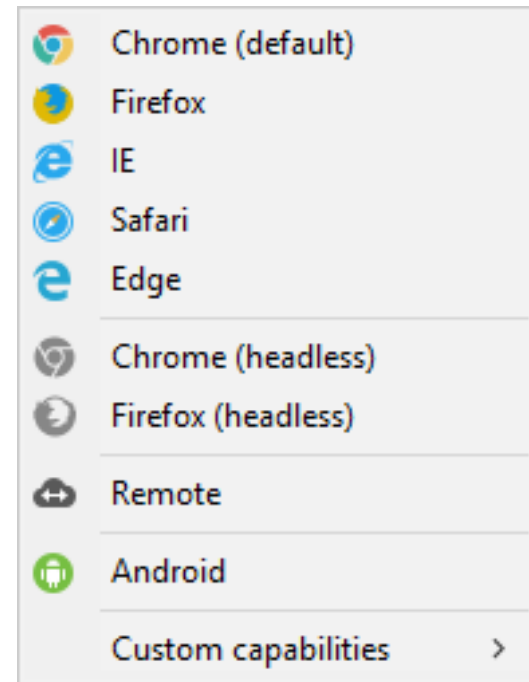
- **Test Maintenance:**

- อัปเดตอัตโนมัติทุกครั้งที่มีการแก้ไข test cases และ suites เมื่อ Test objects มีการเปลี่ยนแปลงเกิดขึ้น
- สะดวกต่อการจัดการและการบำรุงรักษาสคริปต์ ข้อมูลการทดสอบตลอดจน keywords ที่ใช้



Multiple browsers

- สามารถรันการทดสอบได้ทั้งแบบ Local ตลอดจนบริการบน cloud services
- สนับสนุนการทำงานกับเบราว์เซอร์ต่าง ๆ :
 - Chrome
 - Firefox
 - IE
 - Edge
 - Safari
 - Chrome (headless)
 - Firefox (headless)

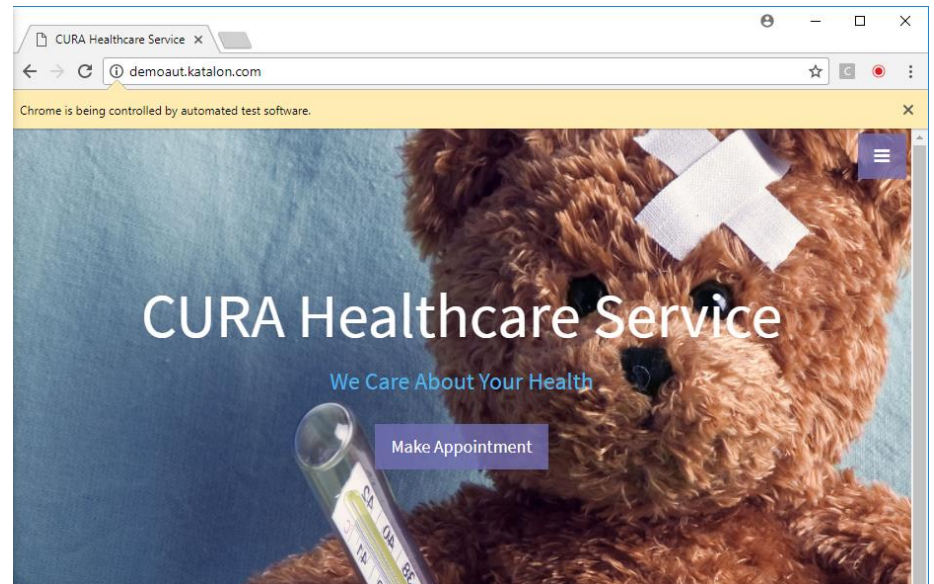
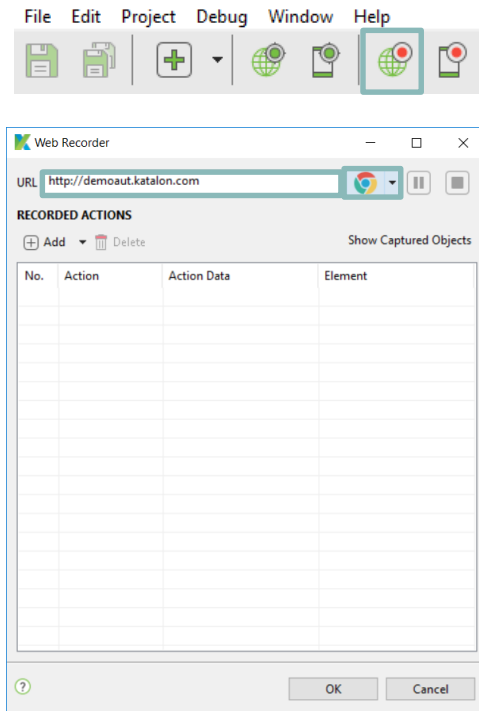


Katalon support

- Katalon Studio แม้ว่าจะเป็นเครื่องมือที่ไม่มีค่าใช้จ่าย แต่มีบริการสนับสนุนแผนงานทางเทคนิคต่าง ๆ หลากหลายราคา ได้แก่
 - Business (ตอบภายในเวลา 24 hours, 1 Helpdesk account, และ 1 support project) – \$2,500 ต่อปี
 - Enterprise (ตอบภายในเวลา 12 hours, 3 Helpdesk accounts, และ 3 support projects) – \$5,000 ต่อปี
 - Enterprise Premium (multiple accounts, exclusive support) – customizable price.

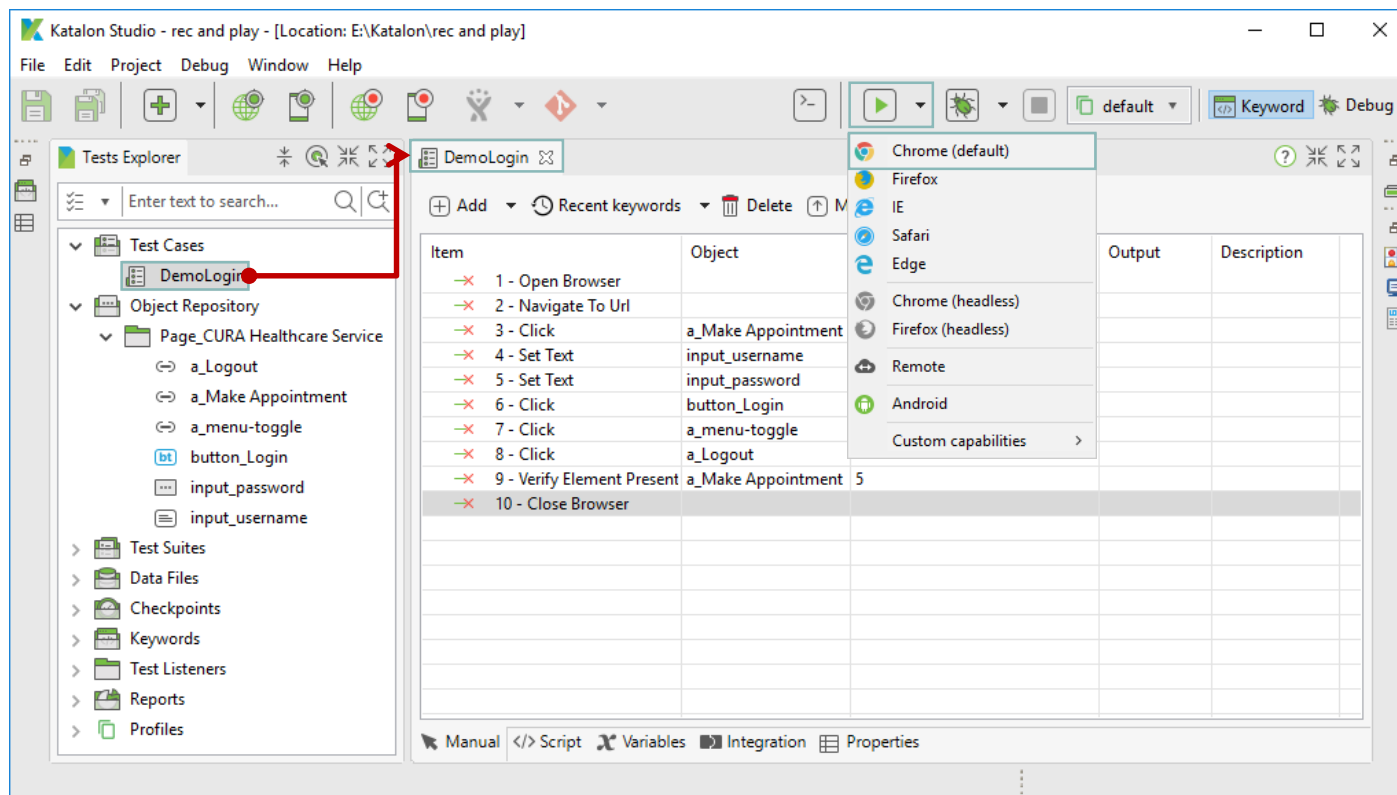
Katalon : Recording

- Katalon Studio สนับสนุนการทำงานแบบ Record การทำงานของ AUT
- เพื่อบันทึกการทำงานของผู้ใช้ร่วมกับระบบในรูปสคริปต์ ซึ่งคุณสมบัตินี้สามารถใช้ได้ทั้งเว็บแอปฯ และ โมบายแอปฯ



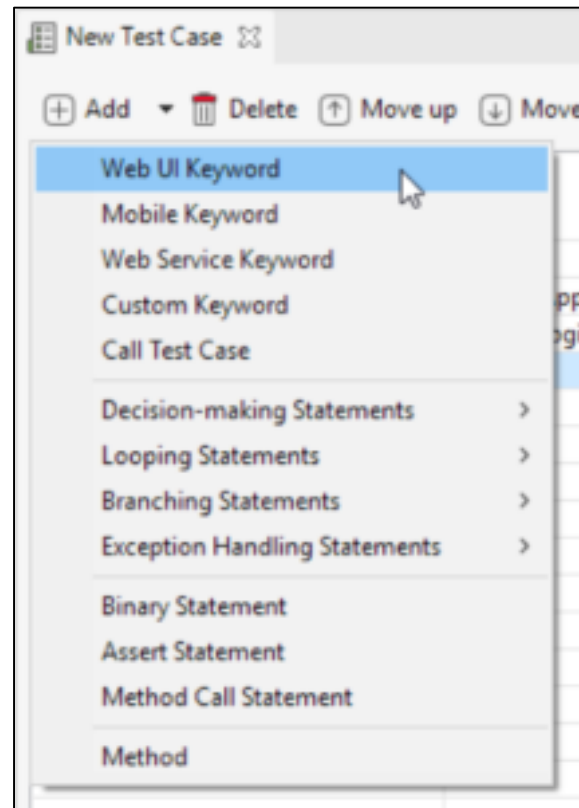
Katalon : Playback

- ส่วนการ Playback เป็นการนำสคริปต์ที่ถูกบันทึกไว้แล้วกลับมารันซ้ำ
- กรณีนี้ผู้ใช้สามารถแก้ไขสคริปต์ตลอดจนค่าตัวแปรต่าง ๆ ได้ตามต้องการ



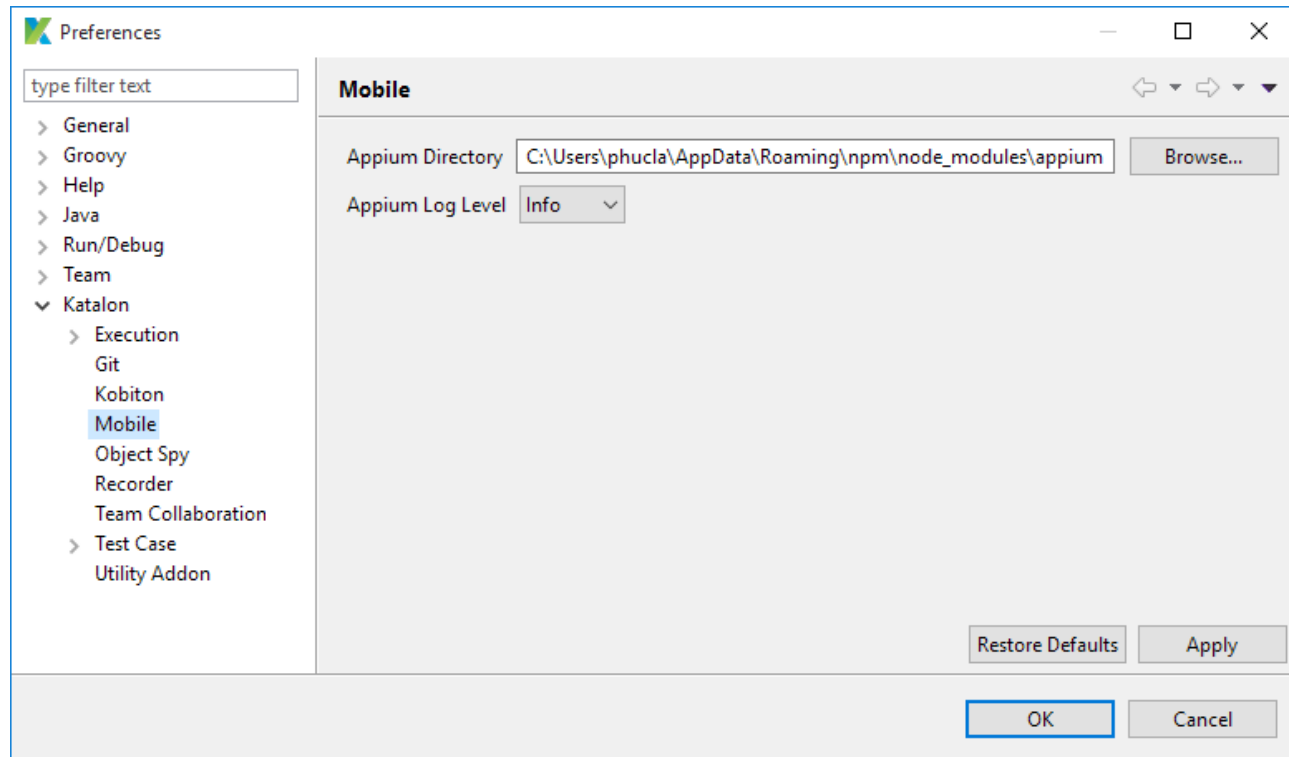
Commands

- Katalon Studio ได้จัดเตรียมคำสั่งต่าง ๆ ไว้เพื่อให้ผู้ใช้สามารถเลือกใช้งานได้อย่างเหมาะสม โดยแยกหมวดหมู่ไว้เพื่อความสะดวกในการเรียกใช้



Setting

- นอกจากนั้นผู้ใช้อย่างยังสามารถตั้งค่าต่าง ๆ ที่จำเป็นได้ทั้งในระดับโปรเจก และระดับของการใช้เครื่องมือทดสอบได้เช่นกัน



Variables

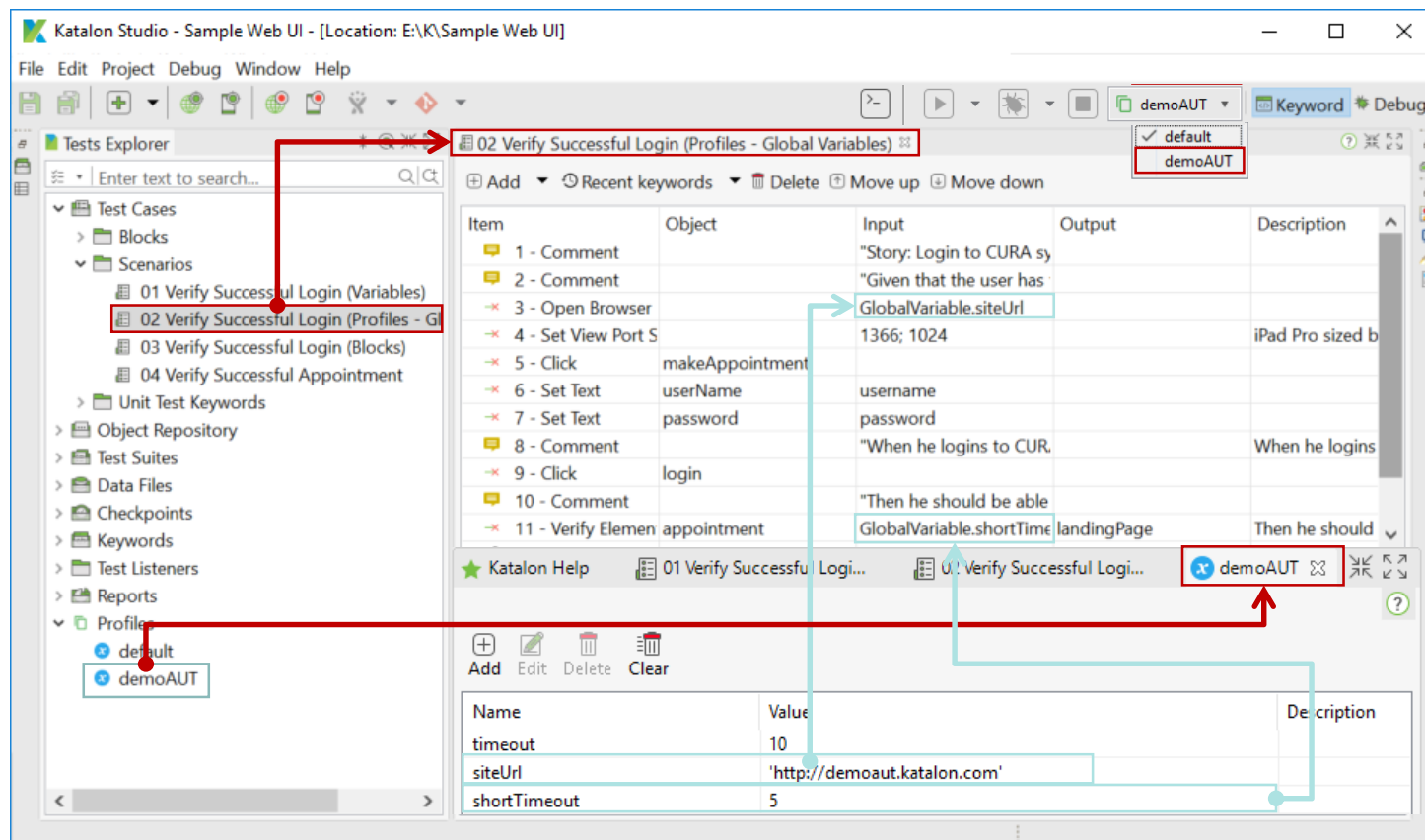
- เพื่อแก้ปัญหาเรื่อง Hard Coded ผู้ใช้สามารถกำหนดค่าตัวแปรเพื่อใช้งานได้ ในหลาย ๆ ระดับ

The screenshot displays the Katalon Studio interface. On the left, the 'Tests Explorer' pane shows a tree view with 'Test Cases' expanded, containing 'Blocks' and 'Scenarios'. Under 'Scenarios', '01 Verify Successful Login (Variables)' is selected. A red arrow points from this selection to the main test case editor. The editor shows a table of test steps for 'Demo Login'. The first table (steps 1-9) includes actions like 'Open Browser', 'Navigate To Url', 'Click', 'Set Text', and 'Close Browser', with inputs like 'http://demoaut.katalon.com/' and 'John Doe'. The second table (steps 10-15) includes actions like 'Click', 'Set Text', 'Click', 'Verify Element Present', 'Method Call Statement', 'Delay', 'Click', and 'Close Browser', with inputs like 'username', 'password', 'login', 'appointment', 'println(landingPage)', and '2'. The bottom status bar shows 'Manual', 'Script', 'Variables', 'Integration', and 'Properties' tabs, with 'Variables' currently active.

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		

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 logs in to CURA system"
8 - Click	login	
9 - Comment		"Then he should be able to login successfully"
10 - Verify Element Present	appointment	5
11 - Method Call Statement		println(landingPage)
12 - Delay		2
13 - Click	menuBar	
14 - Click	logout	
15 - Close Browser		

- ผู้ใช้สามารถกำหนดตัวแปรปกติได้ระดับ Test Case และตัวแปรแบบ Global เพื่อใช้งานในระดับโปรเจกต์ได้เช่นกัน



Creating Scripts

- สร้างสคริปต์สำหรับการทดสอบได้ทั้งแบบ Manual และ Script Mode
- Script Mode ทำงานร่วมกับ Eclipse IDE ที่สนับสนุนโปรแกรมภาษา Groovy รวมไปถึง Java

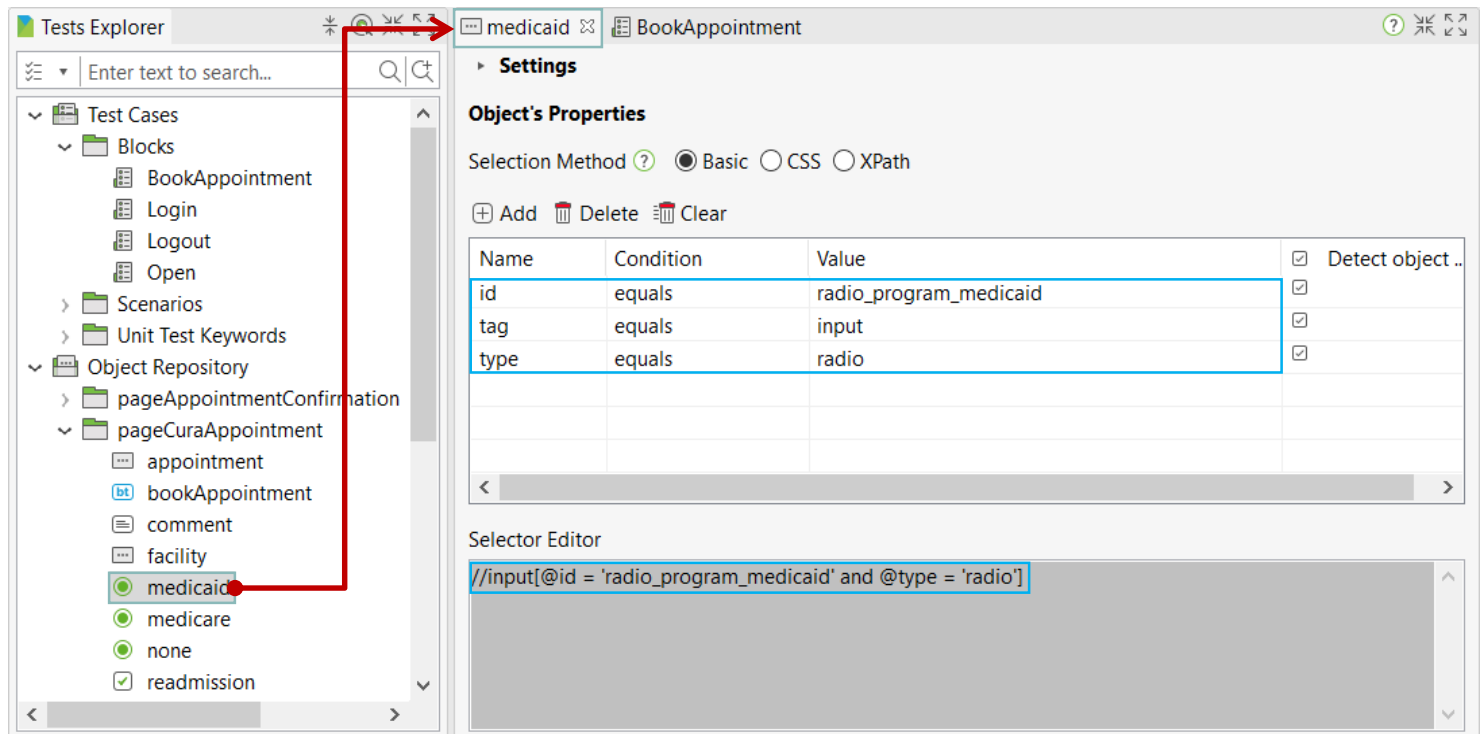
The screenshot displays the Katalon Studio interface with two tabs open. The left tab, titled '01 Verify Successful Login (Variables)', shows a Groovy script for logging into the CURA system. The right tab, titled '02 Verify Successful Login (Profiles - Global Variables)', shows a table representing a test case.

```
import static com.kms.katalon.core.checkpoint.Checkpoint.  
WebUI.comment('Story: Login to CURA system')  
WebUI.comment('Given that the user has the valid login :  
WebUI.openBrowser('http://demoaut.katalon.com')  
WebUI.click(findTestObject('pageCuraHomepage/makeAppoin  
WebUI.setText(findTestObject('pageLogin/userName'), use  
WebUI.setText(findTestObject('pageLogin/password'), pas  
WebUI.comment('When he logs in to CURA system')  
WebUI.click(findTestObject('pageLogin/login'))  
WebUI.comment('Then he should be able to login successfi  
def landingPage = WebUI.verifyElementPresent(findTestOb  
println landingPage  
WebUI.closeBrowser()
```

Item	Object	Input
1 - Comment		"Story: Login to CURA s
2 - Comment		"Given that the user has
3 - Open Browser		GlobalVariable.siteUrl
4 - Click	makeAppointment	
5 - Set Text	userName	username
6 - Set Text	password	password
7 - Comment		"When he logs in to CUF
8 - Click	login	
9 - Comment		"Then he should be abl
10 - Verify Element Present	appointment	GlobalVariable.shortTim
11 - Close Browser		

Object Repository

- ใช้กลไกการจัดเก็บคุณสมบัติของ Test Object ไว้ในกลไกที่เรียกว่า Object Repository เช่นเดียวกับ QTP/UFT



The screenshot displays the Selenium IDE interface. On the left, the 'Tests Explorer' pane shows a tree view of test cases and keywords. The 'Object Repository' is expanded, showing a list of objects. A red arrow points from the 'medicaid' object in the list to the 'medicaid' tab in the main panel. The main panel shows the 'Settings' for the 'medicaid' object, including 'Object's Properties' and a table of attributes.

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

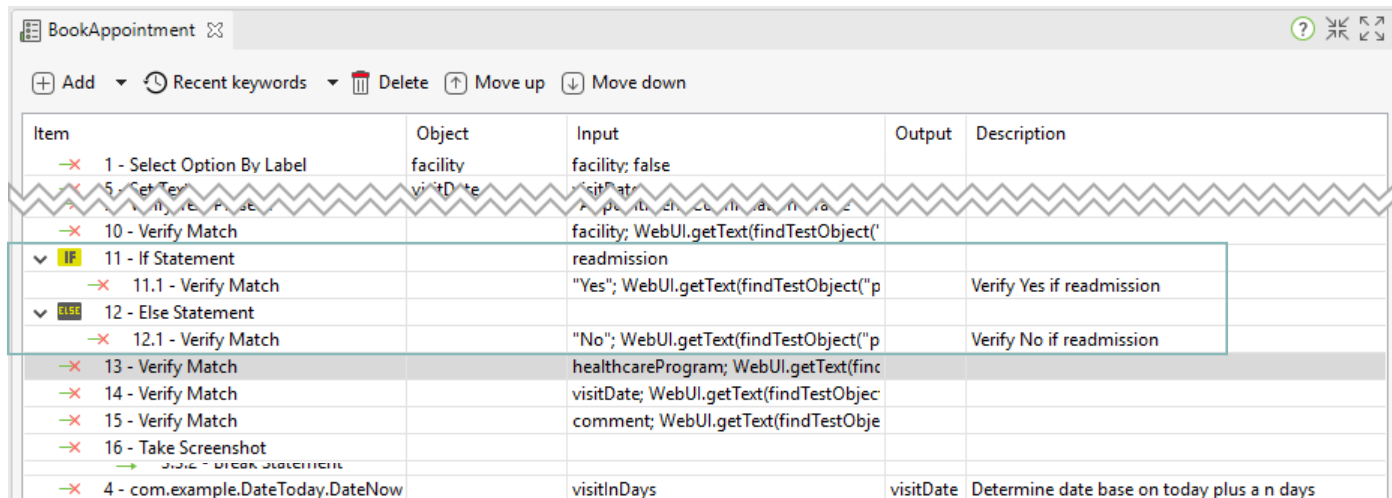
```
//input[@id = 'radio_program_medicaid' and @type = 'radio']
```

Statements

- นอกจากนั้น Katalon Studio ยังสนับสนุนการทำงานประเภท Control Flow ต่าง ๆ ดังนี้
 - Decision-making statement
 - Looping statements
 - Branching statements
 - Exception handling statements

IF Else Statements

- ในการทดสอบอัตโนมัติมักมีการแก้ไขสคริปต์เพื่อให้สามารถทำงานได้ตามความต้องการของผู้ใช้ Katalon Studio จึงได้พัฒนากลไกการทำงานแบบ If-else เพื่อเพิ่มความสะดวกให้กับผู้ใช้



Item	Object	Input	Output	Description
1 - Select Option By Label	facility	facility; false		
5 - Set Text	visitDate	visitDate		
10 - Verify Match		facility; WebUI.getText(findTestObject('pageAppointmentConfirmation/hospitalReadmission'))		
11 - If Statement		readmission		
11.1 - Verify Match		"Yes"; WebUI.getText(findTestObject('pageAppointmentConfirmation/hospitalReadmission'))		Verify Yes if readmission
12 - Else Statement				
12.1 - Verify Match		"No"; WebUI.getText(findTestObject('pageAppointmentConfirmation/hospitalReadmission'))		Verify No if readmission
13 - Verify Match		healthcareProgram; WebUI.getText(findTestObject('pageAppointmentConfirmation/healthcareProgram'))		
14 - Verify Match		visitDate; WebUI.getText(findTestObject('pageAppointmentConfirmation/visitDate'))		
15 - Verify Match		comment; WebUI.getText(findTestObject('pageAppointmentConfirmation/comment'))		
16 - Take Screenshot				
4 - com.example.DateToday.DateNow		visitInDays	visitDate	Determine date base on today plus a n days

```
if (readmission) {  
    'Verify Yes if readmission'  
    WebUI.verifyMatch('Yes', WebUI.getText(findTestObject('pageAppointmentConfirmation/hospitalReadmission')), false)  
} else {  
    'Verify No if readmission'  
    WebUI.verifyMatch('No', WebUI.getText(findTestObject('pageAppointmentConfirmation/hospitalReadmission')), false)  
}
```

Switch Case Statements

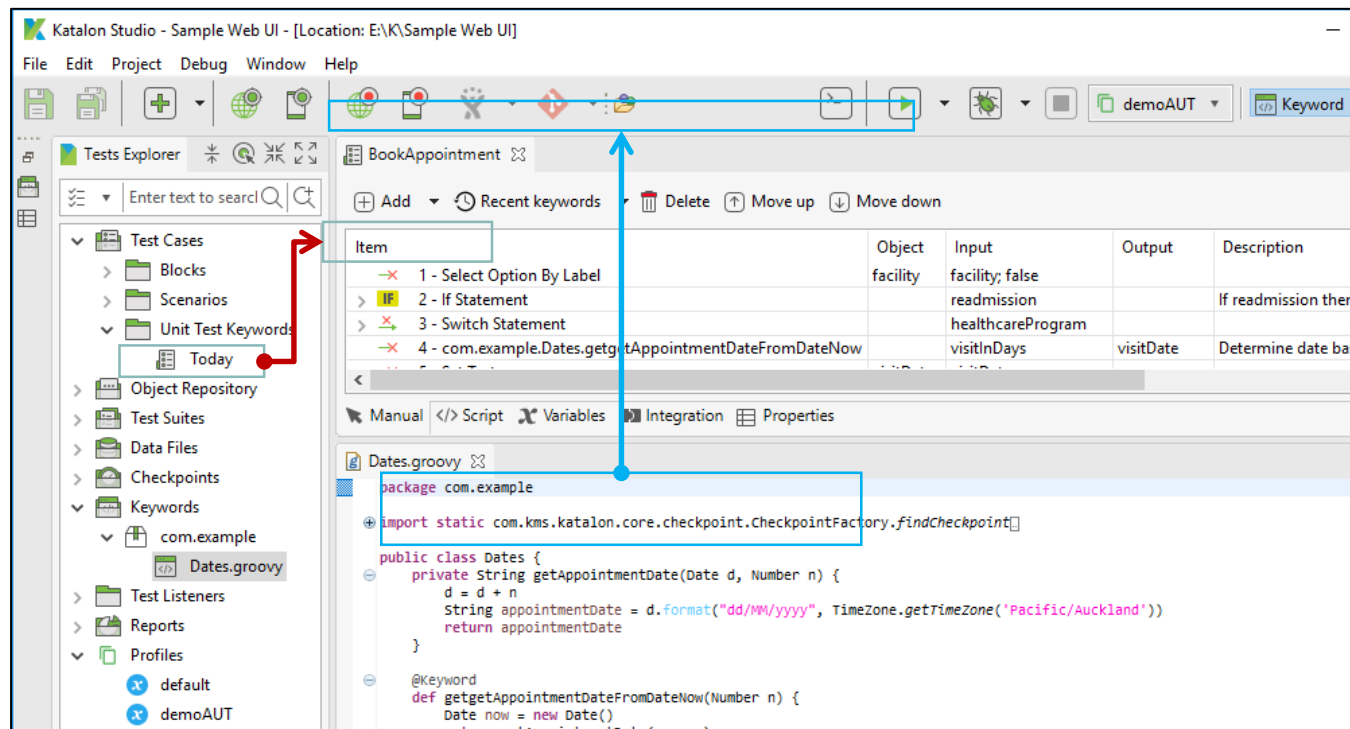
BookAppointment				
+ Add Recent keywords Delete Move up Move down				
Item	Object	Input	Output	Description
1 - Select Option By Label	facility	facility; false		
2 - If Statement		readmission		If readmission then tick readmission
3 - Switch Statement		healthcareProgram		
3.1 - Case Statement		case healthcareProgram = "Medicare":		
3.1.1 - Check	medicare			Case of Medicare then check medicare radio button
3.1.2 - Break Statement				
3.2 - Case Statement		case healthcareProgram = "Medicaid":		
3.2.1 - Check	medicaid			Case of Medicare then check medicaid radio button
3.2.2 - Break Statement				
3.3 - Default Statement				
3.3.1 - Check	none			Check none radio button as default
3.3.2 - Break Statement				
4 - com.example.DateToday.DateNow		visitInDays	visitDate	Determine date base on today plus a n days

Healthcare Program ☒ Medicare ☐ Medicaid ☐ None

```
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  
    default:  
        'Check none radio button as default'  
        WebUI.check(findTestObject('pageCuraAppointment/none'))  
  
        break  
}
```

Keywords

- ในกรณีที่ไม่มีคำสั่ง Built in Command ที่เหมาะสม ผู้ใช้สามารถสร้างชุดคำสั่งใหม่ได้ตามต้องการ โดยใช้กลไก Keywords ที่ Katalon Studio ได้จัดเตรียมไว้แล้ว



Web Service

- นอกจากการทดสอบเว็บแอปฯ แล้ว Katalon Studio ยังสามารถใช้ทดสอบ API ที่ใช้ร่วมกับเว็บเซอร์วิสได้เช่นกัน

The screenshot displays the Katalon Studio interface for configuring and running a REST API test case. The left sidebar shows the 'Tests Explorer' with a tree view containing 'Test Cases', 'Object Repository', 'Test Suites', 'Data Files', 'Checkpoints', 'Keywords', 'Test Listeners', 'Reports', and 'Profiles'. A red arrow points from the 'Test Cases' folder to the 'TC01_REST_Verify Email from list of Comments' test case. A blue arrow points from the 'GET REST_CommentDetails' test case in the 'Object Repository' to the 'GET REST_CommentDetails' test case in the main editor. The main editor shows the test case configuration for 'GET REST_CommentDetails'. The 'Item' table lists the steps: '1 - Send Request' and '2 - Verify Element Property'. The 'Object' column for '1 - Send Request' is 'REST_CommentDetails'. The 'Input' column for '2 - Verify Element Property' is 'response; "[0].email"; "Eliseo@gardner.biz"'. The 'Output' column for '2 - Verify Element Property' is 'response'. The 'Description' column for '2 - Verify Element Property' is 'Verify if comment's email after ser'. The 'Parameters' section shows a table with 'Name' and 'Value' columns. The 'Authorization' section shows 'HTTP Header' and 'HTTP Body' tabs. The 'Type' section shows 'No Authorization' and 'Update to HTTP Header' buttons. The 'Response' section shows the status '200 OK', elapsed time '807 ms', and size '2 KB'. The 'Body' tab is selected, showing the response body in 'pretty' format. The response body is a JSON object:

```
{  "postId": 1,  "id": 1,  "name": "id labore ex et quam laborum",  "email": "Eliseo@gardner.biz",  "body": "laudantium enim quasi est quidem magnam voluptate ipsam eos\ntempora quo necessitatibus\ndolor quam autem quasi\nreiciendis et nam sapiente accusantium"}
```

Item	Object	Input	Output	Description
1 - Send Request	REST_CommentDetails		response	Send a SOAP request and returns i
2 - Verify Element Property		response; "[0].email"; "Eliseo@gardner.biz"	response	Verify if comment's email after ser

Name	Value

Authorization: HTTP Header HTTP Body

Type: No Authorization

Update to HTTP Header

Response: Status: 200 OK Elapsed: 807 ms Size: 2 KB

Body Header

pretty raw preview

```
1 {
2   {
3     "postId": 1,
4     "id": 1,
5     "name": "id labore ex et quam laborum",
6     "email": "Eliseo@gardner.biz",
7     "body": "laudantium enim quasi est quidem magnam voluptate ipsam eos\ntempora quo necessitatibus\ndolor quam autem quasi\nreiciendis et nam sapiente accusantium"
8   },
9   {
10    "postId": 1,
```

Mobile

- นอกจากการทดสอบเว็บแอปฯ แล้ว Katalon Studio ยังสามารถใช้ทดสอบ โมบายแอปฯ ได้เช่นกัน

